



Universidad de Buenos Aires

Facultad de Ingeniería

Propuesta de Trabajo Profesional

Qin – Cluster

Barrabino, Diego	dbarrabino@gmail.com	80183
Moreyra, Martín	moreyramj@gmail.com	84394
http://code.google.com/p/qin-cluster/		

Tutor: Licenciada Adriana Echeverría

Observaciones:

Índice

01.- Introducción.....	3
01.1.- Objetivo	3
01.2.- Alcance	3
01.2.1.- Requerimientos.....	3
01.2.1.1.- Ambiente	3
01.2.1.2.- Aplicación de prueba.....	3
01.2.2.- Software	4
01.2.3.- Hardware.....	4
02.- Motivación del Proyecto	5
03.- Breve reseña de Clustering.....	6
04.- Herramientas de desarrollo.....	8
04.1.- Software.....	8
04.1.1.- Herramientas de desarrollo	8
04.1.2.- Ambientes de desarrollo.....	9
04.1.3.- JBoss	10
04.1.4.- Terracotta	11
04.1.5.- Tomcat	12
04.2.- Hardware	13
05.- Aplicación web de ejemplo	14
05.1.- Casos de uso de la aplicación web	15
06.- Validación de Resultados	16
07.- Metodología.....	17
08.- Calendario	18
09.- Currículo autores.....	20
09.1.- Barrabino, Diego	20
09.2.- Moreyra, Martín	24
10.- Referencias	26

01.- Introducción

01.1.- Objetivo

El propósito del presente Trabajo Profesional, es analizar y comparar dos tipos de arquitecturas para el desarrollo de una Aplicación Web focalizada en brindar escalabilidad, alta disponibilidad y alto desempeño, junto con la estructuración, configuración, administración y despliegue de un entorno productivo con soporte de Clustering. Particularmente, se desean comparar las arquitecturas a desarrollar teniendo en cuenta la utilización de las herramientas Terracotta y JBoss.

01.2.- Alcance

01.2.1.- Requerimientos

01.2.1.1.- Ambiente

Los dos ambientes a considerar, es decir tanto el ambiente armado a partir de JBoss como el ambiente armado a partir de Terracotta, deberán ofrecer:

- Alta disponibilidad
- Escalabilidad
- Alto desempeño

01.2.1.2.- Aplicación de prueba

Ambos entornos descritos anteriormente serán evaluados y validados mediante la interacción con una aplicación web de prueba, la misma será en cuanto a la funcionalidad la misma al probar el entorno montado con JBoss como el entorno montado con Terracotta, con la arquitectura adecuada a cada entorno, el cual es condicionado por cada herramienta.

El detalle de la aplicación se encuentra en la sección Aplicación web de ejemplo.

01.2.2.- Software

La configuración de software mínima común para ejecutar ambos entornos es:

- Cualquier sistema operativo.
- JDK y JRE 1.6u22.
- Dependencias propias de la aplicación de prueba.
- Servidor MySQL 5.1.53, en la computadora determinada para tal fin según configuración de red. El sistema operativo del servidor de base de datos no necesariamente debe ser el mismo que el usado para deployar la aplicación, pero nada impide que sea el mismo.

El software necesario para ejecutar el entorno con JBoss es:

- Configuración de software mínima común.
- JBoss AS 5.1.0 debidamente instalado.

El software necesario para ejecutar el entorno con Terracotta es:

- Configuración de software mínima común.
- Apache Tomcat 6.0.29 debidamente instalado.
- Librerías de Terracotta 3.4.0 disponibles para el proyecto deployado en Apache Tomcat 6.0.29.

Se concluye que los desarrollos que se efectuarán para cumplir con el presente trabajo no tendrán en sí limitaciones de software, y correrán en toda computadora cuya configuración sea compatible con Java, JBoss y Apache Tomcat.

La misma aplicación se prueba se adaptará tanto para el escenario con JBoss como para el escenario con Terracotta, de manera tal que se pueda trabajar con dos ambientes con una diferencia tecnológica clara.

01.2.3.- Hardware

El hardware necesario para ejecutar cada uno de los entornos que se van a preparar para cumplir con el presente trabajo es:

- Cualquier procesador simple core como por ejemplo Pentium IV, o superior.
- Memoria RAM de 1 Gb, o superior.
- Espacio libre de al menos 500 Mb en disco rígido.
- Placa de red o conectividad WIFI.

02.- Motivación del Proyecto

A medida que van apareciendo más aplicaciones web integrales que resuelven cada vez más y más aspectos del negocio que intentan abarcar, más se va dependiendo de dichas soluciones, y más impacto tienen las caídas de los sistemas. Por lo cual, solamente tener una muy buena aplicación web, algo para nada fácil de lograr por cierto, es solo una parte del desafío si se quiere desarrollar una aplicación realmente exitosa, otra parte del desafío es lograr que esté disponible prácticamente siempre.

Una de las soluciones posibles para intentar que las aplicaciones estén online casi bajo cualquier condición, y particularmente la solución que nos interesa, es implementar Clustering en el entorno productivo de la aplicación. De esta forma, por las características propias de la técnica, se puede de mantener la aplicación online incluso en caso de que alguna de las computadoras que conforman el Cluster deje de funcionar, entre otros inconvenientes que podrían surgir.

Por otro lado, vemos cómo a nivel laboral también aumenta paulatinamente el interés por la técnica en cuestión, lo que nos lleva a concluir que dominar la técnica constituye una manera más de defenderse y destacarse a nivel laboral, en un mercado muy dinámico y competitivo.

Revisando las tecnologías disponibles, vemos que podríamos montar un ambiente de tales características usando herramientas que no nos son enteramente ajenas, como ser el servidor JBoss, o una librería de Java denominada Terracotta, entre otras herramientas finalmente descartadas como el servidor Glassfish. También notamos que las herramientas elegidas proponen dos maneras de abordar la implementación de un ambiente de Clustering.

JBoss propone replicar partes de una aplicación en varios servidores sincronizándolos mediante mecanismos provistos por el mismo JBoss, pero esto requiere efectuar implementaciones específicas requeridas por el servidor de aplicaciones para poder realizar esta sincronización, condicionando el diseño de la aplicación.

Terracotta propone montar una aplicación pensada para un servidor y desplegarla en todos los servidores de manera que no tengan conocimiento de existencia entre si. El encargado de la sincronización es justamente esta misma librería, la cual no requiere implementaciones específicas para esta tarea, por lo que en teoría se puede replicar cualquier aplicación.

Ambas maneras son muy atractivas e interesantes, incluso una es prácticamente opuesta a la otra, y se podrá implementar una u otra según las necesidades del caso en una situación del mundo laboral. Finalmente, elegimos las herramientas mencionadas porque nos permiten abordar el área de interés con variantes, lo que consideramos que favorecerá a la riqueza de nuestra investigación.

En consecuencia, es muy interesante para nosotros poder llegar a dominar la técnica de Clusterización, y su implementación de dos maneras distintas, ya que es una técnica que nos llama la atención, y que podríamos aplicar a nivel laboral, con herramientas de fácil acceso.

03.- Breve reseña de Clustering

El término Cluster se aplica a los conjuntos o conglomerados de computadoras contruidos mediante la utilización de componentes de hardware comunes y que se comportan como si fuesen una única computadora.

Hoy en día desempeñan un papel importante en la solución de problemas de las ciencias, las ingenierías y del comercio moderno.

La tecnología de Clusters ha evolucionado en apoyo de actividades que van desde aplicaciones de supercómputo y software de misiones críticas, servidores web y comercio electrónico, hasta bases de datos de alto rendimiento, entre otros usos.

El cómputo con Clusters surge como resultado de la convergencia de varias tendencias actuales que incluyen la disponibilidad de microprocesadores económicos de alto rendimiento y redes de alta velocidad, el desarrollo de herramientas de software para cómputo distribuido de alto rendimiento, así como la creciente necesidad de potencia computacional para aplicaciones que la requieran.

Simplemente, un Cluster es un grupo de múltiples ordenadores unidos mediante una red de alta velocidad, de tal forma que el conjunto es visto como un único ordenador, más potente que los comunes de escritorio.

Los Clusters son usualmente empleados para mejorar el rendimiento y/o la disponibilidad por encima de la que es provista por un solo computador típicamente siendo más económico que computadores individuales de rapidez y disponibilidad comparables.

De un Cluster se espera que presente combinaciones de los siguientes servicios:

- 1.- Alto rendimiento
- 2.- Alta disponibilidad
- 3.- Balanceo de carga
- 4.- Escalabilidad

La construcción de los ordenadores del Cluster es más fácil y económica debido a su flexibilidad: pueden tener todos la misma configuración de hardware y sistema operativo (Cluster homogéneo), diferente rendimiento pero con arquitecturas y sistemas operativos similares (Cluster semi-homogéneo), o tener diferente hardware y sistema operativo (Cluster heterogéneo), lo que hace más fácil y económica su construcción.

Teniendo en cuenta las computadoras disponibles, se concluye que el tipo de Cluster que se implementará un Cluster heterogéneo.

Para que un Cluster funcione como tal, no basta solo con conectar entre sí los ordenadores, sino que es necesario proveer un sistema de manejo del Cluster, el cual se encargue de interactuar con el usuario y los procesos que corren en él para optimizar el funcionamiento.

Las aplicaciones paralelas escalables requieren: buen rendimiento, baja latencia, comunicaciones que dispongan de gran ancho de banda, redes escalables y acceso rápido a archivos. Un Cluster puede satisfacer estos requerimientos usando los recursos que tiene asociados a él.

La tecnología Cluster permite a las organizaciones incrementar su capacidad de procesamiento usando tecnología estándar, tanto en componentes de hardware como de software que pueden adquirirse a un costo relativamente bajo.

04.- Herramientas de desarrollo

04.1.- Software

04.1.1.- Herramientas de desarrollo

El proyecto será desarrollado tal como implica la tecnología ya mencionada, con software open source:

- Ambiente de desarrollo integrado Eclipse Helios para J2EE con plugins de Web Tools Platform (<http://www.eclipse.org/downloads/packages/eclipse-ide-java-developers/heliossr1>).
- Lenguaje de desarrollo Java (JDK y JRE 1.6u22 https://cds.sun.com/is-bin/INTERSHOP.enfinity/WFS/CDS-CDS_Developer-Site/en_US/-/USD/ViewProductDetail-Start?ProductRef=jdk-6u22-oth-JPR@CDS-CDS_Developer).
- Servidor MySQL 5.1.53 (<http://www.mysql.com/downloads/mysql/>).
- Servidor JBoss AS 5.1.0 (<http://sourceforge.net/projects/jboss/files/JBoss/JBoss-5.1.0.GA>), o Apache Tomcat 6.0.29 (<http://tomcat.apache.org/download-60.cgi>) con librerías de Terracotta 3.4.0 (<http://www.terracotta.org/dl/oss-sign-up>) según el caso a desarrollar.
- Motor de persistencia Hibernate 3.6.0 (<http://www.hibernate.org/downloads>).
- Componentes para MVC de Spring Framework 3.0.5 (<http://www.springsource.org/download>).
- JSP – JavaScript; las versiones soportadas tanto por los navegadores más comunes como por los servidores utilizados.
- Apache JMeter 2.4 (<http://jakarta.apache.org/jmeter/>).
- Librerías auxiliares como JodaTime (<http://joda-time.sourceforge.net/>) para manejo de fechas y slf4j (<http://www.slf4j.org/>) para loggear, entre otros menos notables.
- SVN (<http://code.google.com/p/qin-cluster/>) y sus respectivos clientes, ya sea los instalables a modo de plugin en Eclipse como más centrados en el sistema operativo como RapidSVN (<http://rapidsvn.tigris.org/>) o TortoiseSVN (<http://tortoisesvn.tigris.org/>).
- Maven 3.0 (<http://maven.apache.org/>) para administrar el desarrollo de la aplicación de prueba.

Se trabajará con estas herramientas indistintamente sobre Ubuntu Linux o Windows XP o 7, según qué computadora o qué tipo de booteo se decida hacer para cada caso.

04.1.2.- Ambientes de desarrollo

Los ambientes de desarrollo son ambientes compatibles con la tecnología principal que es el lenguaje de programación Java, por lo cual son completamente funcionales a los efectos de desarrollar.

Particularmente, se cuenta con una computadora desktop clon con dual boot entre Windows XP y Ubuntu Linux Desktop 10.10, una computadora netbook Acer Aspire One con dual boot entre Windows XP y Ubuntu Linux Netbook 10.10, y ...

Llegado el caso, si es que se necesita tener más sistemas operativos activos que los que se pueden tener al mismo tiempo con las computadoras disponibles, se ejecutará en cualquier computadora las instancias que sean necesarias del software de virtualización VirtualBox 3.2.10 (<http://www.virtualbox.org/>), sobre el que se puede correr cualquier Linux o Windows disponible. En una primera aproximación, los sistemas operativos virtualizados no escaparán a Ubuntu Desktop 10.10, Windows 7 o Windows XP.

04.1.3.- JBoss

JBoss es un servidor de aplicaciones J2EE de código abierto implementado en Java puro. Al estar basado en Java, JBoss puede ser utilizado en cualquier sistema operativo para el que esté disponible Java. Los principales desarrolladores trabajan para una empresa de servicios, JBoss Inc., adquirida por Red Hat en abril del 2006, fundada por Marc Fleury, el creador de la primera versión de JBoss. El proyecto está apoyado por una red mundial de colaboradores. Los ingresos de la empresa están basados en un modelo de negocio de servicios.

JBoss implementa todo el paquete de servicios de J2EE.

JBoss soporta Clusterización de manera nativa. Simplemente, al levantar los servidores que conforman el Cluster, se ejecuta un comando que los sincroniza y logra que se comporten en conjunto como si fueran un solo servidor.

04.1.4.- Terracotta

Terracotta es un software de código abierto que permite crear aplicaciones Java que pueden escalar a cualquier cantidad de computadoras, sin tener que crear código adicional o usar bases de datos para compartir datos dentro del Cluster.

Terracotta utiliza el concepto de Memoria Adjunta a la Red (NAM - Network Attached Memory). El uso de NAM le permite a Terracotta distribuir en Cluster a Máquinas Virtuales Java (JVM) directamente debajo de las aplicaciones, brindando alta disponibilidad y escalabilidad de forma transparente.

La ventaja de Terracotta es que la aplicación preparada para funcionar en Cluster es exactamente igual a una aplicación Java común. Todos los conceptos y librerías que se usan (POJOs, Spring, Hibernate, threads, sincronización, etc.) funcionan de la misma manera con Terracotta en un entorno distribuido, de la misma manera que funcionan en una única máquina virtual con muchos hilos de ejecución.

Terracotta funciona a nivel de memoria, por lo que no es necesario heredar de ninguna clase ni implementar ninguna interfaz para compatir objetos entre todas las máquinas virtuales del Cluster (ni siquiera es necesario implementar `java.io.Serializable`).

En definitiva, se puede programar la aplicación de manera natural, y se deja a Terracotta que administre todo el trabajo para crear un entorno escalable y de alta disponibilidad.

04.1.5.- Tomcat

Tomcat es un Servlet Container, el cual provee el entorno de ejecución de servlets, mediante los cuales se puede programar una aplicación web.

Los mencionados servlets son simples clases de Java, que reciben pedidos HTTP y envían respuestas que son mostradas en el navegador web del cliente.

Si bien Tomcat y Terracotta son herramientas totalmente diferentes, inclusive de diferentes compañías, éstas se complementan, ya que se puede desarrollar una aplicación web al estilo tradicional utilizando Tomcat y sin tener en cuenta aspectos de la Clusterización, y luego utilizar Terracotta para replicar esta aplicación web y lograr un comportamiento de Cluster.

04.2- Hardware

Se dispone de 3 computadoras de prueba, una desktop, una netbook y una notebook. La computadora desktop será usada ampliamente al momento de investigar y desarrollar, y las portátiles no solamente participarán del desarrollo e investigación, sino que además también serán las computadoras con que se presentará el trabajo.

Las mismas son:

Desktop: computadora clon, procesador Intel(R) Core(TM)2 Quad CPU Q8200 @ 2.33GHz, placa madre Intel DG41TY versión AAE47335-202, placa de red Realtek RTL8168D(P)/8111D(P) PCI-E Gigabit Ethernet NIC, memoria RAM de 2 plaquetas de 2 Gb DDR2-800 (400 MHz) SDRAM, disco rígido principal SAMSUNG HD501LJ 500 GBytes IDE SATA-II, disco rígido secundario SAMSUNG HD120IJ 120 GBytes IDE SATA-II, sistemas operativos Windows XP y Ubuntu Linux Desktop 10.10 en dual boot; en ambos sistemas operativos ya existe otro Ubuntu Linux Desktop 10.10 y un Windows 7 virtualizados con VirtualBox 3.2.10.

Netbook: Acer Aspire One (http://www.acer.com/aspireone/aspireone_8_9/), procesador Intel(R) Atom(TM) CPU N270 @ 1.60GHz, placa madre Acer, placa de red Realtek RTL8102E/RTL8103E Family PCI-E Fast Ethernet NIC, placa de red wifi Atheros AR5007EG Wireless Network Adapter, memoria RAM de 2 plaquetas de 512 Mb DDR2 (PC2-5300) 333 MHz (DDR2 667), disco rígido Hitachi HTS543216L9A300 160.0 GB 5400 RPM Serial ATA, sistemas operativos Windows XP y Ubuntu Linux Netbook 10.10 en dual boot

Notebook: Hewlett-Packard Presario F500 (GM639LA#AC8) WLAN Broadcom 802.11b/g - Controladora de red NVIDIA nForce, memoria RAM de 2 plaquetas de 512 Mb DDR2(PC2-5300) 333 MHz (DDR2 667), disco rígido 120GB SATA-150, sistemas operativos Windows XP y Ubuntu Linux en dual boot

05.- Aplicación web de ejemplo

La Aplicación Web que se considera para efectuar el análisis propuesto en el presente trabajo, es una aplicación administrativa para recibir trabajos y evaluar el desempeño académico de los alumnos de un instituto educativo de gran tamaño.

El escenario que se considera es el siguiente:

- 1.- Se provee un usuario del sistema a cada uno de los alumnos y a cada uno de los profesores, quienes luego incorporan su contraseña privada de acceso.
- 2.- Todos los alumnos están obligados, periódicamente, a incorporar al sistema todos los trabajos prácticos que van realizando a medida que transcurre el ciclo lectivo, junto con otra información pertinente que permite clasificar los trabajos prácticos de manera correcta.
- 3.- En el instituto hay una gran cantidad de alumnos, y se espera que en la mayor parte de las materias se les solicite muchos trabajos prácticos, los cuales pueden llegar a ser considerablemente grandes. El uso diario del sistema es muy exigente, ya que todos los días existe un muy alto tráfico de trabajos, considerando todas las materias en todos los distintos niveles educativos que ofrece la institución.
- 4.- Los profesores deben estar constantemente evaluando el material cargado, ya que necesitan dicho material para determinar la condición de regularidad de los alumnos, para calificar los trabajos prácticos, y para que a fin de año se puedan obtener estadísticas generales del desempeño del alumnado. Los profesores deben terminar sus evaluaciones, ya sea de los trabajos o la general a fin de año, en determinadas fechas. Por lo cual cabe señalar que todos los días, algunos profesores deben obligatoriamente tener preparadas ciertas evaluaciones.
- 5.- El sistema debe estar siempre disponible, ya que si así no lo hiciera, los alumnos se verían impedidos de entregar a tiempo sus trabajos, lo que pondría en peligro su regularidad, al mismo tiempo que gran cantidad de profesores se verían impedidos de terminar sus evaluaciones a tiempo. El sistema debe ser escalable porque la dirigencia del instituto planea seguir expandiendo sus actividades, lo cual indefectiblemente hará crecer la aplicación y creará más presión sobre los servidores. El sistema debe mostrar un altísimo desempeño, para poder sobrellevar que muchos alumnos incorporen sus trabajos al mismo tiempo, muchos profesores los consulten al mismo tiempo, que mientras que los alumnos cargan, los profesores evalúen, etc.

Se propone para el presente trabajo profesional, considerar la implementación de una primera versión mínima del sistema explicado, que permita a los alumnos cargar un mínimo de información, que permita a los profesores hacer una evaluación mínima de dicha información, y que permita en el futuro incorporar nuevos módulos como ser el módulo de generación de prioridades para inscripción, nuevas estadísticas generales, etc.. Esta implementación debe necesariamente cumplir con los requisitos no funcionales de escalabilidad, desempeño, y disponibilidad.

05.1.- Casos de uso de la aplicación web

Elegimos un conjunto de casos de uso que pondrán a prueba las características deseadas de alta disponibilidad y buen desempeño, el cual se logra a través del Clustering.

Asumimos que el sistema debe ser utilizado por toda una universidad con gran cantidad de alumnos, por lo que la carga introducida por cualquiera de los casos mencionados a continuación sera muy grande.

- 1.- Log in: proveer acceso al sistema mediante usuarios que podrán tener acceso a las pantallas de carga o evaluación según corresponda
- 2.- Subir trabajos prácticos: pantallas de carga de los trabajos prácticos de los alumnos.
- 3.- Inscripción: pantalla de inscripción de alumnos
- 4.- Búsqueda de TPs: pantalla de búsqueda de trabajos según cuatrimestre y materia.
- 5.- Visualización de TPs: pantalla de visualización de los trabajos cargados.
- 6.- Evaluación de TPs: pantalla para la calificación de los trabajos cargados por parte de los profesores.
- 7.- Seguimiento de materias en tiempo real: pantalla donde se mostrará dada una materia, la cantidad de TPs cargados, y se graficará la cantidad de TPs según su calificación, actualizando el gráfico cada 5 segundos.
- 8.- Estadística de inscriptos según ciudad: pantalla donde se mostrará la cantidad de alumnos inscriptos por ciudad.

06.- Validación de Resultados

Para concluir el análisis de ambas arquitecturas, se desarrollaran casos de prueba sobre la aplicación web para ambos entornos (JBoss y Terracotta), los cuales se focalizarán en los requerimientos de rendimiento y disponibilidad.

En ambos entornos se ejecutaran pruebas de carga automatizadas, estas pruebas consistirán en fijar un número determinado de usuarios concurrentes que realizarán determinadas acciones en el sistema. Para esto utilizaremos la herramienta JMeter, con la cual podemos aparte de automatizar estas pruebas, ver resultados de carga vs tiempo de respuesta. Con esta información podremos realizar comparaciones sobre la aplicaciones web desplegada en cada herramienta.

Por otro lado también se realizarán pruebas manuales para verificar la alta disponibilidad, estas pruebas consistirán en la baja de algún servidor, y la verificación de que al usuario no se le presente ningún problema para seguir conectado al sistema.

Los casos mencionados serán especificados al finalizar la etapa de análisis, para poder tener un conocimiento mucho mayor relacionado a las problemáticas del Clustering a la hora de desarrollar los mismos.

07.- Metodología

La metodología elegida para llevar adelante este proyecto, consiste en definir cuáles son los objetivos del mismo, y a partir de dichos objetivos, organizar el tiempo en secciones evolutivas del proyecto, a saber:

- 1.- Concepción (4 semanas de duración)
- 2.- Elaboración (4 semanas de duración)
- 3.- Análisis de la arquitectura (12 semanas de duración)
- 4.- Construcción (8 semanas de duración)
- 5.- Finalización (4 semanas de duración)

Por la naturaleza del proyecto, las etapas más importantes son las primeras tres etapas.

A lo largo de todo el proyecto, se administrará un informe general e integrador, y particularmente sobre el fin de cada sección se desarrollarán los informes correspondientes que den cuenta que se llevó a cabo dicha etapa, qué se hizo, que se esperaba conseguir, qué se consiguió, etc., según la naturaleza de cada sección. Cada una de estos informes se adosará al informe general e integrador.

Se tiene más detalle de esto en la sección Calendario de este informe.

08.- Calendario

Distinguimos en este proyecto las siguientes etapas:

- 1.- Concepción (4 semanas de duración)
 - Investigación y análisis de las herramientas disponibles que proveen Clustering
 - Selección de las herramientas que aseguren diversidad técnica, para enriquecer la investigación ; en este caso como ya se aclaró, JBoss y Terracotta
 - Análisis y elección de en qué tipo de escenario es aplicable la tecnología elegida, lo cual puede verse en la sección "*Aplicación web de ejemplo*"
- 2.- Elaboración (4 semanas de duración)
 - Creación del repositorio público (<http://code.google.com/p/qin-cluster/>).
 - Confección de la propuesta de trabajo profesional
 - Definición de los casos de uso de la aplicación de prueba
- 3.- Análisis de la arquitectura (12 semanas de duración)
 - Análisis del escenario implementado con Terracotta, denominado "*escenario 1*"
 - Análisis del escenario implementado con JBoss, denominado "*escenario 2*"
 - Desarrollo de casos de prueba generales acordes a un entorno que soporte Clustering (ver sección "Validación de resultados")
 - Desarrollo de los casos de prueba particulares para el escenario 1, si corresponde
 - Desarrollo de los casos de prueba particulares para el escenario 2, si corresponde
 - Entrega de documentación reflejando las características de esta etapa, a incluir en el informe general final
- 4.- Construcción (8 semanas de duración)
 - Implementación del escenario 1: esta tarea implica montar y configurar el escenario que se despliega con Terracotta, sin aún considerar la aplicación de prueba.
 - Desarrollo de la aplicación de prueba para el escenario 1
 - Implementación del escenario 2: esta tarea implica montar y configurar el escenario que se despliega con JBoss, sin aún considerar la aplicación de prueba.
 - Desarrollo de la aplicación de prueba para el escenario 2
 - Ejecución de las pruebas generales, y recolección de información
 - Ejecución de las pruebas particulares del escenario 1, y recolección de información
 - Ejecución de las pruebas particulares del escenario 2, y recolección de información
 - Entrega de documentación reflejando las características de esta etapa, a incluir en el informe general final

- Comparación de los resultados obtenidos
- Comparación de los resultados obtenidos
- Documentación de las conclusiones del proyecto
- Informe general final

Concepción				Elaboración				Análisis de la arquitectura												Construcción								Finalización			
1	2	3	4	1	2	3	4	1	2	3	4	5	6	7	8	9	10	11	12	1	2	3	4	5	6	7	8	1	2	3	4
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
Semanas																															

09.- Currículo autores

09.1.- Barrabino, Diego

Datos personales:

- Apellido y nombre: BARRABINO, Diego.
- Documento Nacional de Identidad (D.N.I.): 29.064.455.
- Fecha de nacimiento: 7 de septiembre de 1981.
- Edad: 29 años.
- Domicilio: Quito 4247 1º 3, CPA C1212ABM, entre Mármol y Muñiz, Almagro, Ciudad Autónoma de Buenos Aires.
- Teléfono particular: 011 – 3528 – 6888.
- Casilla de e-mail: dbarrabino@gmail.com.

Antecedentes laborales:

a).- Certant Technology Solutions (Av. De Mayo 666, 4º piso – Capital Federal. Teléfono: 5219-0855/6)

. Pasantía de seis meses (diciembre 2003 – junio 2004) en la citada empresa de informática. Se realizaron las siguientes actividades, en las oficinas de la mencionada empresa tanto como en las instalaciones del cliente: programación, mantenimiento, análisis de proyectos, análisis de programas y documentación de proyectos.

Durante los últimos tres meses de los seis citados, presté servicios en UOL – Sinectis, en el departamento de Sistemas, área de Tecnología, sita en Florida 537 6º piso (1005), donde se efectuaron tareas de mantenimiento variadas, programación en Perl, tanto para desarrollar nuevos programas o scripts como para mantener y supervisar programas o scripts anteriores o de otros programadores, HTML y consultas básicas a bases de datos en SQL.

b).- Dirección de Informática – Gerencia de Organización y Sistemas- CONICET (Av. Rivadavia 1906, 2º piso, oficina 13 – Capital Federal. Teléfono: 5983 – 1420, interno 601)

. Pasantía desde el primero de octubre de 2004 hasta abril de 2005, bajo contrato desde entonces.

Conformo un grupo de trabajo de aproximadamente 22 personas directamente bajo supervisión del director de Informática del CONICET, cuyo objetivo es desarrollar, implementar, relevar, mantener y proponer aplicaciones informáticas de diverso porte y amplia funcionalidad según el caso, siempre alineadas con los objetivos estratégicos del CONICET. Las herramientas utilizadas son Java, Eclipse, JbuilderX, EnterpriseArchitect, MySQL Workbench, Jira, Tomcat, Glassfish, Hibernate, EclipseLink, Struts1 y 2, JSP, JavaScript, MySQL, Oracle, HTML, etc. Actualmente los proyectos más notorios son el sistema SIGEVA (Sistema Integral de Gestión y Evaluación), la implantación como proyecto del sistema SIGEVA en otras instituciones, el sistema DFD que provee de Firma Digital (implementación completa; primer caso en el Estado Argentino) al sistema SIGEVA, el sistema SIGERH (Sistema Integral de Gestión de Recursos Humanos) con su implementación de Credenciales de identificación del personal, el sistema de Datawarehouse, el naciente sistema CONICET Digital, entre otros, como ser el sistema CER de Certificaciones de Servicio, el sistema MEC de Mesa de Entrada del CONICET, etc. Con el correr del tiempo, he participado incluso desde sus orígenes en todos los proyectos.

En el grupo mencionado, ejecuto tareas de relevamiento, análisis, diseño, desarrollo, implementación, control y seguimiento de diversas funcionalidades de los sistemas administrados, en un marco de colaboración total y recíproca con absolutamente todos mis compañeros, varios de los cuales conformamos un núcleo estable desde hace varios años.

Dentro del grupo a modo global, actualmente me encuentro dirigiendo un subgrupo enfocado principalmente en el proyecto SIGERH, y en segundo término aunque adquiriendo cada vez más relevancia, en una próxima implantación de SIGEVA.

Antecedentes académicos:

CBC (año 2000)

- Álgebra (código 27). Curso 42708. Nota: 7. Primer cuatrimestre de 2000.
- Introducción al Pensamiento Científico (código 40). Curso 44001. Nota: 9. Primer cuatrimestre de 2000.
- Física (código 03). Curso 40302. Nota: 7. Primer cuatrimestre de 2000.
- Análisis Matemático 1 (código 28). Curso 42808. Nota: 8. Segundo cuatrimestre de 2000.
- Sociedad y Estado (código 24). Curso 42422. Nota: 8. Segundo cuatrimestre de 2000.
- Química (código 05). Curso 40504. Nota: 9. Segundo cuatrimestre de 2000.

Carrera 10 – Ingeniería en Informática (año 2001 a la fecha – padrón 80183)

- Algoritmos y Programación I (programación en Pascal) (código 7540). Cátedra: López - Vega. Nota: 9 (11/07/2001 L85 F43). Primer cuatrimestre de 2001.
- Física I A (código 6201). Cátedra: Menikheim. Nota: 7 (31/07/2001 L100 F210). Final revisado por la prof. Lombardo. Primer cuatrimestre de 2001.
- Análisis Matemático II A (código 6103). Cátedra: Patetta. Nota: 10 (11/12/2001 L148 F52). Segundo cuatrimestre de 2001.
- Química (código 6301). Cátedra: Kim – Roble. Nota: 8 (19/12/2001 L68 F217). Segundo cuatrimestre de 2001.
- Álgebra II A (código 6108). Cátedra: Álvarez Juliá. Nota: 6 (27/12/2001 L146 F143). Segundo cuatrimestre de 2001.
- Algoritmos y Programación II (programación en Pascal) (código 7541). Cátedra: Mandrafina. Nota: 7 (06/03/2002 L86 F133). Segundo cuatrimestre de 2001.
- Algoritmos y Programación III (programación en Delphi) (código 7507). Cátedra: Fíntela. Nota: 9 (16/07/2002 L87 F2). Primer cuatrimestre de 2002.
- Análisis Matemático III A (código 6110). Cátedra: Murmis. Nota: 5 (19/07/2002 L143 F212). Primer cuatrimestre de 2002.
- Probabilidad y Estadística B (código 6109). Cátedra: Busch. Nota: 4 (19/12/2002 L147 F124). Final revisado por el prof. Sacerdotti. Segundo cuatrimestre de 2002.
- Análisis Numérico I (código 7512). Cátedra: Tarela - Cavaliere. Nota: 7 (16/12/2002 L87 F205). Segundo cuatrimestre de 2002.

- Física II A (código 6203). Cátedra: Cremaschi – Mesaros – Hogart. Nota: 7 (18/02/2003 L102 F33). Final revisado por el prof. Leone. Segundo cuatrimestre de 2002.
- Laboratorio (código 6602). Cátedra: Bertuccio – Reiser. Nota: 5 (18/07/2003 L128 F214).
Primer cuatrimestre de 2003.
- Física III D (código 6215). Cátedra: Arcondo. Nota: 8 (08/07/2003 L102 F100). Primer cuatrimestre de 2003.
- Estructura del Computador (código 6670). Cátedra: M. C. Ginzburg. Nota: 9 (15/08/2003 L129 F51). Primer cuatrimestre del 2003.
- Simulación (programación en GPSS; código 7526). Cátedra: Nota: 4 (17/12/2003 L89 F175). Segundo cuatrimestre de 2003.
- Organización de Computadoras (código 6620). Cátedra: Hamkalo. Nota: 10 (11/12/2003 L129 F96). Segundo cuatrimestre de 2003.
- Taller de Programación I (código 7542). Cátedra: Veiga. Nota: 8 (26/07/2005 L92 F238). Primer cuatrimestre de 2005.
- Organización de Datos (código 7506). Cátedra: Saubidet. Nota: 7 (11/12/2006 L95 F104). Segundo cuatrimestre de 2006.
- Sistemas Operativos (código 7508). Cátedra: Clúa. Nota 10 (05/07/2007 L96 F146). Primer cuatrimestre de 2007.
- Análisis de la información (código 7509). Cátedra: González. Nota 5 (10/07/2007 L96 F163). Primer cuatrimestre de 2007.
- Estructura de las Organizaciones (código 7112). Cátedra: Barmack. Nota: 8 (27/12/2007 L145 F207). Segundo cuatrimestre de 2007.
- Modelos y Optimización I (código 7114). Cátedra: Ramonet. Nota: 8 (26/12/2007 L145 F199). Segundo cuatrimestre de 2007.
- Técnicas de Diseño (código 7510). Cátedra: Pantaleo. Nota: 8 (07/07/2008 L98 F186). Primer cuatrimestre de 2008.
- Información en las Organizaciones (código 7113). Cátedra: Zambrano. Nota: 6 (10/07/2008 L146 F149). Primer cuatrimestre de 2008.
- Base de Datos (código 7515). Cátedra: Alé. Nota: 5 (07/10/2009 L102 F84). Segundo cuatrimestre de 2008.
- Introducción a los Sistemas Distribuidos (código 7543). Cátedra: Hamelin. Nota: 5 (23/02/2010 L103 F82). Segundo cuatrimestre de 2008.
- Taller de Programación II (código 7552). Cátedra: Servetto. Nota: 10 (05/03/2010 L103 F176). Primer cuatrimestre de 2009.
- Administración y Control de Proyectos Informáticos I (código 7544). Cátedra: Martínez. Nota: 7 (28/09/2009 L102 F45). Primer cuatrimestre de 2009.
- Taller de Desarrollo de Proyectos I (código 7545). Cátedra: Pineiro – Pignataro. Nota: 9 (15/02/2010 L103 F31). Segundo cuatrimestre de 2009.
- Legislación y Ejercicio Profesional de la Ingeniería en Informática (código 7140). Cátedra: Papaleo – Noremberg. Nota: 7 (22/12/2009 L149 F120). Segundo cuatrimestre de 2009.
- Taller de Desarrollo de Proyectos II (código 7547). Cátedra: Fontela. Nota: 9 (06/07/2010 L104 F13). Primer cuatrimestre de 2010.
- Administración y Control de Proyectos Informáticos II (código 7546). Cátedra: Martínez. Nota: 6 (27/07/2010 L104 F153). Primer cuatrimestre de 2010.

Actualmente

Cursando Calidad en el Desarrollo de Sistemas (código 7548), cátedra Pantaleo, e Introducción a los Sistemas Inteligentes (código 7550), cátedra Ochoa.

Total a la fecha:

CBC: 6 materias, promedio 8

Carrera 10 – Ingeniería en Informática: 32 materias, promedio 7,28

Total: 38 materias, promedio 7,39

Suma de créditos: 192

09.2.- Moreyra, Martín

Datos personales:

- Apellido y nombre: Moreyra, Martín Jorge.
- Documento Nacional de Identidad (D.N.I.): 30.892.909.
- Fecha de nacimiento: 6 de junio de 1984.
- Edad: 26 años.
- Domicilio: Ciudad de la Paz 3272 – 4/A, CP 1429, Ciudad Autónoma de Buenos Aires.
- Teléfono particular: 011 – 5339-8528.
- Casilla de e-mail: moreyramj@gmail.com.

Antecedentes laborales:

- a) Globant- (Febrero 2006 – Junio 2009): Participación en diferentes proyectos como desarrollador: red social (FunkySexyCool), sistema de búsqueda y compra de vuelos (Orbitz), sistema de administración de información sobre vuelos (OAG). Herramientas utilizadas: Java, JSP, Velocoty, Struts, Spring, Hibernate, MySQL, Tomcat.
- b) ITR- (Julio 2009 - Diciembre 2009): Proyecto para Swiss Medical, Parte del grupo de desarrollo del sistema que maneja historiales médicos, internaciones y otros. Las herramientas utilizadas fueron: J2EE, Servlets, Sybase portal y Sybase DB.
- c) Swiss Medical- (Enero 2010 – Septiembre 2010): idem b)
- d) OSPIM- (Agosto 2010 – Presente): Formo parte de un grupo de desarrolladores quienes estamos trabajando sobre el nuevo sistema interno de la obra social. El sistema comprende diferentes áreas, afiliaciones, contabilidad, tesorería, entre otros. Mis tareas comprenden desde el relevamiento de los requerimientos, su documentación, diseño y desarrollo. Las herramientas utilizadas son: Java, Servlets, Struts, Liferay, JSP, Postgres, Tomcat.

Antecedentes académicos:

CBC (año 2003)

- Álgebra (código 27)
- Introducción al Pensamiento Científico (código 40). Curso 44001.
- Física (código 03). Curso 40302.
- Análisis Matemático 1 (código 28). Curso 42808.
- Sociedad y Estado (código 24). Curso 42422.
- Química (código 05). Curso 40504.

10 – Ingeniería en Informática (año 2004 a la fecha – padrón 84394)

- Física I A (código 6201) Nota: 6 (10-08-2004 L103 F127)
- Algoritmos Y Programacion I (código 7540) Nota: 6 (13-08-2004 L91 F109)
- Algebra II A (código 6108) Nota: 5 (09-12-2004 L150 F65)

- Algoritmos Y Programacion II (código 7541) Nota: 9 (17-12-2004 L91 F188)
- Analisis Matematico II A (código 6103) Nota: 4 (22-02-2005 L151 F16)
- Quimica (código 6301) Nota: 6 (14-07-2005 L71 F218)
- Algoritmos Y Programacion III (código 7507) Nota: 7 (08-08-2005 L93 F37)
- Analisis Numerico I (código 7512) Nota: 7 (12-08-2005 L93 F60)
- Fisica II A (código 6203) Nota: 5 (15-12-2005 L104 F98)
- Probabilidad Y Estadistica B (código 6109) Nota: 5 (03-03-2006 L152 F59)
- Estructura Del Computador (código 6670) Nota: 8 (13-07-2006 L132 F207)
- Laboratorio (código 6602) Nota: 6 (13-07-2006 L132 F217)
- Fisica III D (código 6215) Nota: 6 (R 08-08-2006 L105 F33)
- Organizacion De Datos (código 7506) Nota: 8 (19-07-2007 L96 F206)
- Sistemas Operativos (código 7508) Nota: 10 (27-12-2007 L97 F202)
- Analisis Matematico III A (código 6110) Nota: 6 (25-02-2008 L149 F124)
- Taller De Programacion I (código 7542) Nota: 9 (08-07-2008 L98 F202)
- Estructura De Las Organizaciones (código 7112) Nota: 4 (30-07-2008 L147 F5)
- Organización De Computadoras (código 6620) Nota: 6 (14-08-2008 L136 F79)
- Analisis De La Informacion (código 7509) Nota: 6 (15-12-2008 L100 F11)
- Modelos Y Optimizacion I (código 7114) Nota: 8 (16-02-2009 L147 F230)
- Informacion En Las Organizaciones (código 7113) Nota: 4 (24-02-2009 L148 F31)
- Simulacion (código 7526) Nota: 10 (05-08-2009 L101 F106)
- Tecnicas De Diseño (código 7510) Nota: 7 (10-08-2009 L101 F133)
- Taller De Programacion II (Código 7552) Nota: 10 (21-08-2009 L101 F219)
- Taller De Desarrollo De Proyectos I (Código 7545) Nota: 9 (08-02-2010 L102 F238)
- Introduccion A Los Sistemas Inteligentes (Código 7550) Nota: 6 (22-02-2010 L103 F88)
- Introduccion A Los Sistemas Distribuidos (Código 7543) Nota: 7 (23-02-2010 L103 F82)
- Base De Datos (Código 7515) Nota: 7 (21-07-2010 L104 F114)

Actualmente

Cursando Calidad en el Desarrollo de Sistemas (código 7548), cátedra Pantaleo, y Administración y Control de Proyectos Informáticos (código 7544).

Total a la fecha:

CBC: 6 materias

Carrera 10 – Ingeniería en Informática: 29 materias, Promedio 6,8

Total: 35 materias

Suma de créditos: 176

10.- Referencias

Terracotta:

<http://www.terracotta.org/>

Tomcat:

<http://tomcat.apache.org/>

JBoss:

<http://jboss.org/>

<http://es.wikipedia.org/wiki/JBoss>

Sesiones Web:

<http://www.terracotta.org/web-sessions/?was-expanded=web-sessions-submenu>

Cache distribuida:

<http://www.terracotta.org/ehcache/ehcache-dx>

<http://jboss.org/jboss-cache>

Clusterización:

<http://docs.jboss.org/jbossas/jboss4guide/r4/html/cluster.chapt.html>

<http://www.terracotta.org/platform/how-terracotta-works>

<http://www.terracotta.org/platform/platform-concepts>

<http://www.terracotta.org/platform/hello-clustered-world>

http://es.wikipedia.org/wiki/Cluster_%28inform%C3%A1tica%29

Testing:

<http://jakarta.apache.org/jmeter/>