



Webbot de Datos Bibliométricos (WDB).

Integrantes :

De Domingo Gonzalo, Guastadisegni Federico, Rodríguez Nahuel.

Cátedra :

Proyecto de Software
Licenciatura en Sistemas.

2015

Índice

I. Proceso de selección de un modelo de ciclo de vida

- A) Ciclos de vida disponibles
- B) Modelo de ciclo de vida seleccionado

II. Procesos de Gestión del Proyecto

- A) Introducción
- B) Proceso de iniciación del proyecto
 - I. Plan de gestión del proyecto
 - a) Interesados del proyecto
 - b) Distribución de tareas y diagrama Gantt
 - c) Estimación por Puntos de Función
 - d) Estimación por COCOMO II
 - e) Mapa de actividades
 - C) Procesos de Seguimiento y Control del proyecto
 - 1. Análisis de riesgos
 - 2. Registro Histórico de Proyectos
- D) Proceso de gestión de la Calidad del Software
 - I. Plan de Garantía de Calidad de Software

III. Procesos de predesarrollo

- A) Introducción
- B) Requerimientos del Usuario
 - I. Proceso de exploración de conceptos
 - a) Situación Actual
 - b) Dominio del problema
 - c) Informe preliminar de necesidades
 - d) Soluciones Alternativas
 - e) Solución Recomendada
 - C) Proceso de asignación del sistema

IV. Procesos Orientados al Desarrollo de Software

- A) Introducción
- B) Requisitos de software
- C) Requerimientos del Usuario
- D) Requisitos Funcionales
- E) Requisitos no funcionales
- F) Requisitos en Negativo
- G) Requisitos de sistema
- H) Diagrama de contexto
- I) Diagrama de flujo de datos
- J) Diagrama de caso de uso
- K) Diagrama de paquete de caso de uso
- L) Escenarios de caso de uso
- M) Diagramas de secuencia
- N) Diagrama de clases
- O) Diagrama de entidad-relación

V. Procesos de postdesarrollo

- A) Proceso de mantenimiento
- B) Proceso de retiro

VI. Procesos integrales del proyecto

- A) Procesos de verificación y validación
 - I. Introducción
 - 2. Procesos de verificación y validación
 - 3. Plan de pruebas

VII. Anexo I : Manual de instalación

VIII. Anexo II : Manual de usuario

IX. Anexo III : Ejecución de pruebas

X. Anexo IV : Repositorios utilizados

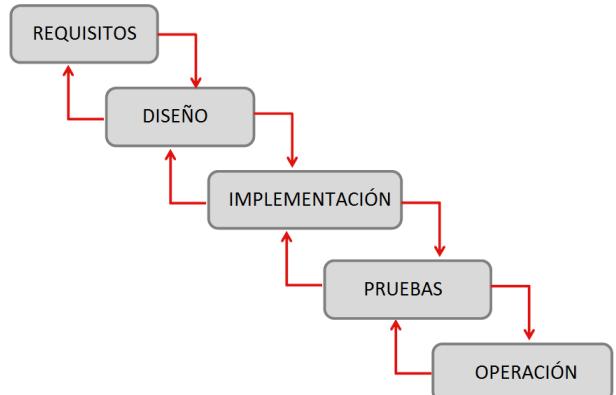
XI. Anexo V: Versionado de documentos

Webbot de Datos Bibliométricos (WDB)

*De Domingo Gonzalo, Guastadisegni Federico, Rodríguez Nahuel
Licenciatura en Sistemas – Departamento de Desarrollo Productivo y Tecnológico
Universidad Nacional de Lanús.*

Resumen-En el presente documento se incluyen todas las actividades realizadas para el proyecto Webbot de Datos Bibliométricos, perteneciente a la cátedra Proyecto de Software de la Carrera Licenciatura en Sistemas, Universidad Nacional de Lanús.

Palabras Clave—*Webbot, Scrapy, Crawler, UNLa, Universidad Nacional de Lanús.*



I. Proceso de selección de un modelo de ciclo de vida

A. Ciclos de vida disponibles

Para realizar nuestro proyecto analizamos los tres ciclos de vida más comunes: en cascada, en espiral y prototipado.

El ciclo de vida en cascada es útil al tener los requisitos bien definidos y congelados en el tiempo.

El ciclo de vida en espiral permite orientar el desarrollo teniendo en cuenta los riesgos y los costos del proyecto. En nuestro caso lo descartamos porque no necesitamos enfocarnos en los riesgos y costos de nuestro proyecto.

El ciclo de vida prototipado es adecuado en dos situaciones :

Cuando los requisitos no estén bien definidos o cambien continuamente y cuando el desarrollador no está seguro si la solución planteada se puede implementar correctamente.

B. Modelo de ciclo de vida seleccionado

Debido a las características del proyecto (requisitos bien definidos y congelados, sin necesidad de análisis de riesgos ni costos y por lo conversado con los clientes), y luego de contemplar los modelos posibles, se eligió el Ciclo de Vida en cascada (ver figura I.B.1) con una particularidad:

La fase de desarrollo se realizará como un prototipo incremental (ver figura I.B.2).

Figura I.B.1 – Ciclo de vida en cascada.

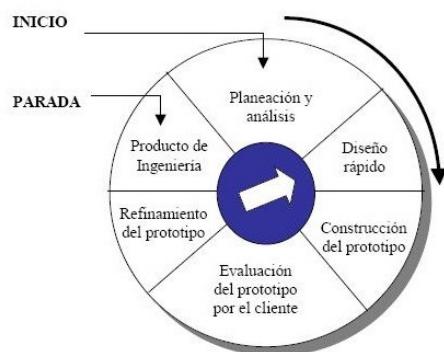


Figura I.B.2 – Ciclo de vida Prototipado incremental pensado para la etapa de desarrollo.

II. Procesos de Gestión del Proyecto

A. Introducción

En esta sección se busca crear la estructura el proyecto y asegurar una adecuada gestión del mismo a lo largo de todo el ciclo de vida del software.

B. Proceso de iniciación del proyecto

Esta sección abarca todas las actividades de creación de la estructura del proyecto.

1. Plan de gestión del proyecto

El plan de gestión consta de una serie de análisis realizados a continuación.

a) Interesados del proyecto

- Documentación: Rodríguez Nahuel.
- Programación: Guastadisegni Federico, De Domingo Gonzalo.

b) Distribución de tareas y diagrama Gantt

La distribución de tareas se especifica en la figura II.B.1, mientras que el diagrama Gantt se puede ver en la figura II.B.2.

c) Estimación por Puntos de Función

Se estimó el tamaño del software mediante el uso de Puntos de Función (Ver Figura II.B.3).

d) Estimación por COCOMO II

Se estimó el esfuerzo del software mediante el uso de COCOMO II (Ver Figura II.B.4).

e) Mapa de actividades

En la tabla II.B.5 se muestra el mapa de actividades seleccionado para el ciclo de vida del proyecto. En la tabla II.B.6 se muestran las referencias.

Id	i	Procesos	duración	Comienzo	Finalización
1		Webbot de datos bibliométricos	75 días	jue 03/09/15	mié 16/12/15
2		Requisitos del sistema global	14 días	jue 03/09/15	mar 22/09/15
3	■■■	Selección de un MCV	2 días	jue 03/09/15	vie 04/09/15
4		Iniciación del proyecto	3 días	lun 07/09/15	mié 09/09/15
5		Exploración de conceptos	3 días	jue 10/09/15	lun 14/09/15
6		Definición requisitos	6 días	mar 15/09/15	mar 22/09/15
7		Diseño Preliminar y Detallado	21 días	mié 23/09/15	mié 21/10/15
8		Iniciación, planificación y estimación	3 días	mie 23/09/15	vie 25/09/15
9		Seguimiento y control	2 días	lun 28/09/15	mar 29/09/15
10		Gestión de la calidad	2 días	mié 30/09/15	jue 01/10/15
11		Verificación y validación	4 días	vie 02/10/15	mié 07/10/15
12		Diseño	10 días	jue 08/10/15	mié 21/10/15
13		Código	20 días	jue 22/10/15	mié 18/11/15
14		Implementación	20 días	jue 22/10/15	mié 18/11/15
15		Pruebas de unidad	4 días	jue 19/11/15	mar 24/11/15
16		Pruebas de unidad	4 días	jue 19/11/15	mar 24/11/15
17		Integración del sistema	7 días	mié 25/11/15	jue 03/12/15
18		Configuración	3 días	mié 25/11/15	vie 27/11/15
19		Integración del sistema	4 días	lun 30/11/15	jue 03/12/15
20		Prueba del sistema de software	3 días	vie 04/12/15	mar 08/12/15
21		Instalación y aceptación	3 días	vie 04/12/15	mar 08/12/15
22		Integración Software - Hardware	6 días	mié 09/12/15	mié 16/12/15
23		Operación y soporte	4 días	mié 09/12/15	lun 14/12/15
24		Entrega documentación final	2 días	mar 15/12/15	mié 16/12/15

Figura II.B.1 – Distribución de tareas.

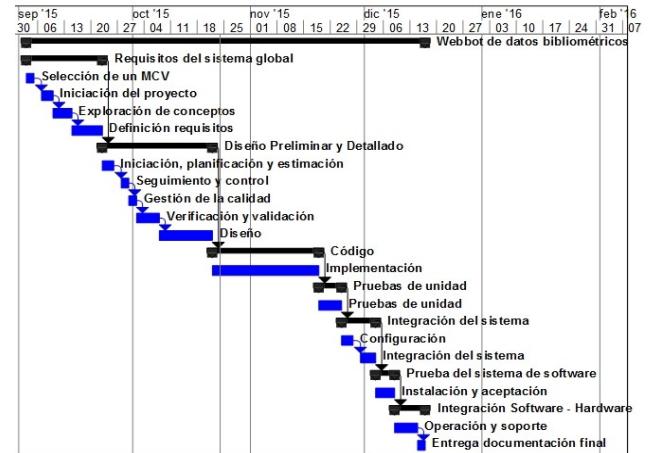


Figura II.B.2 – Diagrama de Gantt.

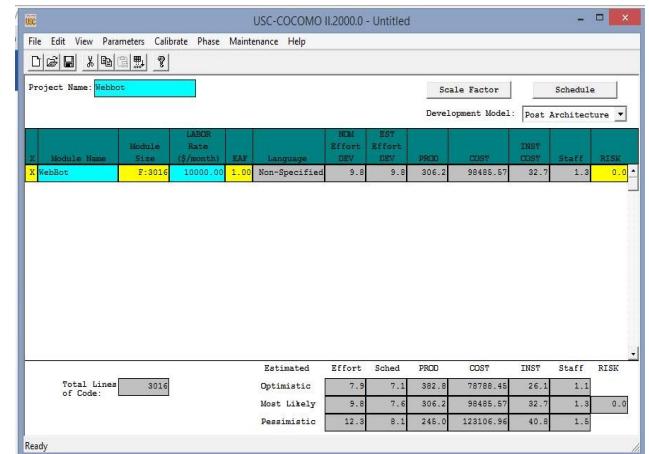


Figura II.B.3 – Estimación mediante Puntos de Función obtenido con USC-COCOMO II.

Descripción	Fórmula	Resultado
Esfuerzo	$E = a * [KLDC]^b$	(Hombre – Mes)
Tiempo	$T = c * E^d$	(Meses)
Personas	$P = E / T$	

Siendo $a = 3,00$; $b = 1,12$; $c = 2,5$; $d = 0,35$

LC Totales	Esfuerzo	Meses	Persona
3000	10,27	5,65	1,82

Estimación	
Personas:	3
Duración del proyecto:	2,65 Meses

Figura II.B.4 – Estimación Mediante COCOMO II (Utilizando el Modelo Semiacoplado para proyectos intermedios).

Tabla II.B.5 – Mapa de actividades.

	A R	D I	C D	O I	P M	O
Actividades de los Procesos						
Proceso de Exploración de Conceptos						
Identificar las ideas o necesidades	X					
Refinar y finalizar la idea o necesidad		X				
Proceso de selección de un MCVS						
Identificar los posibles MCVS	X					
Seleccionar un modelo para el proyecto		X				
Proceso de iniciación, Planificación y Estimación del Proyecto						
Asignar los recursos del proyecto	X	X				
Planificar la gestión del proyecto	X	X				
Proceso de Seguimiento y Control del Proyecto						
Analizar riesgos	X	X				
Realizar la planificación de contingencias	X	X	X	X	X	
Proceso de Gestión de la calidad del Software						
Planificar la garantía de calidad del software	X	X				
Gestionar la calidad del Software	X	X			X	
Proceso de Asignación del sistema						
Analizar las funciones del sistema	X	X	X			
Proceso de Análisis de Requisitos						
Definir los requisitos del software	X					
Priorizar e Integrar los requisitos del software		X	X			
Proceso de Diseño						
Realizar diseño preliminar	X	X				
Diseño de base de datos	X	X				
Desarrollar algoritmos	X	X	X			
Realizar diseño detallado	X					
Reaplicar ciclo de vida (opcional)	X	X	X	X	X	
Proceso de Implementación e Integración						
Crear los datos de prueba	X		X			
Crear el código fuente			X			
Crear la documentación de operación					X	

	A R	D I	C D	O I	P M	O M
Actividades de los Procesos						
Proceso de instalación y Aceptación						
Planificar la instalación						X
Instalar el software						X
Proceso de Operación y Soporte						
Operar el sistema						X
Proveer de asistencia técnica y consultas						X
Proceso de Mantenimiento						
Realizar el mantenimiento correctivo						X
Reaplicar el ciclo de vida del software						X
Proceso de Verificación y Validación						
Planificar la verificación y validación					X	
Ejecutar las tareas de verificación y validación						X
Planificar las pruebas			X	X	X	
Desarrollar las especificaciones de las pruebas			X	X	X	
Ejecutar las pruebas						X
Proceso de Configuración						
Planificar la Gestión de configuración					X	X
Realizar la identificación de la configuración				X		X
Realizar el control de la configuración						X
Realizar informe del estado de la configuración						X
Proceso de documentación						
Planificar la documentación			X	X		
Implementar la documentación			X	X	X	X

Tabla II.B.6 – Referencias.

AR = Análisis de requisitos
DI = Diseño
COD = Codificación.
PI = Pruebas e Integración
OM = Operación y Mantenimiento

C. Procesos de Seguimiento y Control del proyecto

Consiste en un proceso iterativo de seguimiento durante el ciclo de vida del software.

1. Análisis de riesgos

En la tabla II.C.1 se puede visualizar el análisis de riesgo realizado, así como el plan de contingencias.

2. Registro Histórico de Proyectos

Este proceso no se aplica a nuestro proyecto, debido a que nuestro grupo de trabajo no generó proyectos previamente.

#	Descripción	Impacto	Prioridad	Probab.	Acción a seguir
1	Dificultad incorrecta asociada a la resolución del proyecto	Retrasos en la entrega de documentación y en el prototipo	Alta	50%	Mantener la documentación actualizada y validada por los clientes y asignar más tiempo a las tareas más demandantes
2	Cambio del formato de las páginas web, desde las cuales el Webbot obtendrá la información	No se puede continuar con la construcción del prototipo	Alta	25%	Durante la creación del prototipo, verificar una vez por semana que las páginas mantengan el formato adecuado, buscar páginas alternativas
3	Mala planificación	Imposibilidad de terminar el proyecto, de terminarlo fuera de término, mala utilización de recursos	Alta	50%	Mantener el estándar IEEE 1074. Corregir documentación con los clientes.
4	Bugs encontrados en Scrapy, MySql o Solr	Dificulta la integración del sistema	Alta	25%	Se debe renegociar con el cliente y establecer nuevos plazos y objetivos
5	Pérdida de recursos humanos	Retraso y aumento de dificultad del proyecto, malestar en los demás recursos humanos	Alta	25%	Mantener el compromiso con el equipo de trabajo, afianzar relaciones.

Tabla II.C.1 – Análisis de riesgos y plan de contingencias.

D. Proceso de gestión de la Calidad del Software

El objetivo es la planificación y administración de las acciones necesarias para garantizar la calidad del software.

1. Plan de Garantía de Calidad de Software

Se utilizarán herramientas como Scrapy, Django, MySql y Apache Solr, herramientas que proveen seguridad, robustez y facilidad de uso, y que pueden ser adaptadas a cualquier plataforma.

Además, cada una dispone de buena documentación y una comunidad activa, en caso de que se presente algún problema.

Utilizaremos Git, una herramienta que nos va a permitir mantener versiones de la documentación y del código de manera ordenada y segura.

Se realizará la documentación en tiempo y forma para facilitar el mantenimiento del software a futuro.

Se utilizará versionado sobre cada documento correspondiente a cada subprocesso de desarrollo de software. Así, podemos llevar a cabo un control sobre los cambios además de trazabilidad a lo largo del tiempo.

De esa manera, es posible controlar la documentación de línea base. Así, los futuros cambios pueden ser realizados en los documentos correspondientes, dejando intactos los que no sean alcanzados por los cambios realizados.

III. Procesos de predesarrollo

A. Introducción

El planteo del sistema surge a partir de la necesidad de crear un artefacto web que permita la recolección de información acerca de publicaciones artículos científicos que contengan los indicadores bibliométricos solicitados por el cliente.

Un Webbot es una herramienta diseñada para recorrer una cantidad de páginas (definida por el desarrollador) según la cantidad de información que se necesite, para posteriormente devolver la información para ser utilizada en base a las necesidades del proyecto.

En este documento vamos a describir cuáles son los requisitos funcionales que consideramos necesarios para el desarrollo de nuestro artefacto software. Para llegar a ello, previamente realizamos un cuestionario con nuestras inquietudes a nuestros clientes (profesores de la cátedra), así como también mediante las preguntas hechas en clase.

B. Requerimientos del Usuario

Se requiere el desarrollo de un artefacto capaz de contabilizar los indicadores solicitados.

En esta sección se va realizar un análisis de la situación actual y las necesidades que llevan al desarrollo del artefacto software. Se contemplan diferentes soluciones (ver punto 4 más adelante) y se propone la alternativa elegida de común acuerdo con el cliente.

A) Proceso de exploración de conceptos

A continuación se detallan los diferentes procesos referidos a este tema.

1. Situación Actual

Actualmente los Web Bots (WB), o Web Crawlers son utilizados ampliamente para el indexado de páginas web. Esto es útil para páginas que utilizan motores de búsqueda, y que requieren obtener información relacionada a un tema o palabra en específico de una forma rápida, confiable y ordenada.

2. Dominio del problema

El objetivo del proyecto es elaborar un Web Bot (WB), dirigido a recolectar Datos Bibliométricos, almacenar esa información e indexarla con la finalidad de ser utilizada posteriormente. Se utilizarán hasta un máximo de 10 páginas con información bibliométrica, con un formato ya definido.

3. Informe preliminar de necesidades

Se han detectado las siguientes necesidades:

- Entender el objetivo y las limitaciones que tiene un Web Bot (WB) desarrollado en un ambiente universitario.
- Desarrollar la aplicación utilizando el lenguaje Python versión 2.7.x.
- El sistema es un intermediario, cuya información recolectada será utilizada por un usuario.
- El sistema basará sus búsquedas en una "plantilla" predeterminada, que será creada a partir de tags presentes en 5 o 10 páginas que contengan material bibliométrico, en las cuales estarán presentes los indicadores propuestos por el cliente.
- Dependiendo del tipo de página, se estimará un tiempo aproximado de desarrollo.

4. Soluciones Alternativas

El software puede ser desarrollado sin la estructura de un patrón de diseño conocido, o usando algún patrón como por ejemplo MVC (no utilizaremos Vistas, se construirá una página de pruebas).

Pueden utilizarse distintos paradigmas como por ejemplo el Orientado a Objetos, entre otros.

El lenguaje Python es un requisito para el desarrollo del artefacto, por lo tanto se analizaron distintos Frameworks y librerías correspondientes a dicho lenguaje que podrían llegar a ser útiles para el desarrollo del artefacto, como por ejemplo: Django, Scrapy, entre otros.

5. Solución Recomendada

Luego de conversar con el cliente y explorar las distintas alternativas, se decidió de común acuerdo utilizar un patrón de diseño que incluya modelo y controlador, y una página web de “testeo” para nuestro uso en la fase de pruebas.

Por otro lado, se observa en la librería Scrapy un buen soporte para el desarrollo de la aplicación, ofreciendo además las herramientas necesarias para la misma, por lo que se decidió junto al cliente que el mismo sería el más conveniente para el desarrollo de nuestro proyecto.

Se utilizará el ORM (mapeo objeto-relacional) de Django y se integrará junto a Scrapy.

C. Proceso de asignación del sistema

La aplicación no contempla este punto ya que es una aplicación web y no es necesario el desarrollo de Hardware.

IV. Procesos Orientados al Desarrollo de Software

A. Introducción

En esta sección se desarrollarán los documentos de requisitos, así como los distintos diagramas de diseño para el proyecto desarrollado.

B. Requisitos de software

En esta sección se enunciarán los distintos tipos de requerimientos del artefacto software a desarrollar. Hay 3 (tres) tipos de proyectos aplicables:

- Para las páginas “caóticas” (aquellas que no presentan un orden para la información, que no presentan una limitación entre los distintos elementos que componen la página, como el título, copete, texto, imágenes, etc.) se considera un tiempo de trabajo de 5 a 10 días.
- Para las páginas relativamente organizadas se considera un tiempo de trabajo de 3 a 5 días.
- Para las páginas correctamente organizadas y separadas en secciones adecuadas, se considera un tiempo de trabajo de 3 días.

C. Requerimientos del Usuario

Se requiere el desarrollo de un artefacto capaz de contabilizar los indicadores solicitados.

D. Requisitos Funcionales

Aquí, vamos a describir cuál va a ser el comportamiento del artefacto, así como sus requisitos funcionales:

- El sistema deberá ser un software intermediario, ya que la funcionalidad pedida tiene que ser utilizada por otro artefacto software.
- El sistema basará sus búsquedas en una "plantilla" predeterminada, que será creada a partir de tags presentes en 10 páginas que contengan material bibliométrico, en las cuales estarán presentes los indicadores propuestos por el cliente.
- Posteriormente, esa información se indexará con Solr.

E. Requisitos no funcionales

A continuación, se enumeran los requisitos no funcionales, los cuales representan las restricciones sobre los Requisitos Funcionales:

- El sistema se deberá implementar sobre la infraestructura existente en la Universidad Nacional de Lanús.
- No existe posibilidad de utilizar frameworks distintos al solicitado por el cliente (Django). Aunque, se puede complementar el desarrollo del artefacto con librerías que nos sean útiles. La versión de Python a utilizar será la v2.7.
- Se debe trabajar con los indicadores bibliométricos solicitados por el cliente.

F. Requisitos en Negativo

En la siguiente lista se enumeran los Requisitos en negativo, los cuales indican las funciones que no formarán parte del sistema:

- El sistema no contabilizará ni realizará ningún tipo de acción sobre indicadores no especificados por los clientes.
- El usuario no podrá utilizar directamente el Webbot, ni tampoco participará en los procesos internos del artefacto software.
- Durante el desarrollo, no se contempla corregir errores/bugs en las herramientas utilizadas.

G. Requisitos de sistema

A continuación, se detallan los Requisitos de sistema, los cuales representan las funcionalidades que deberá tener el sistema:

- Poder utilizar el lenguaje Python, con el Framework Django, integrando Scrapy y Apache Solr, para poder recolectar y procesar la información requerida.
- La recolección de información debe terminar en un momento. La misma no puede durar indefinidamente.
- La búsqueda y almacenamiento debe funcionar dentro de parámetros normales de tiempo. Consideramos como parámetro normal a un período de tiempo de 20 a 25 minutos contados a partir de la ejecución del Webbot. El límite de funcionamiento depende de la cantidad de publicaciones requeridas por el usuario, así como también la cantidad de publicaciones que contenga la página.
- El sistema será una aplicación Web.

H. Diagrama de contexto

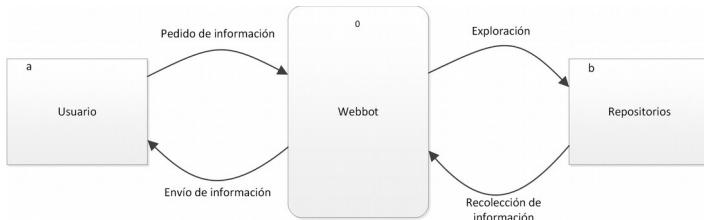


Figura IV.H.1 – Diagrama de contexto.

I. Diagrama de flujo de datos

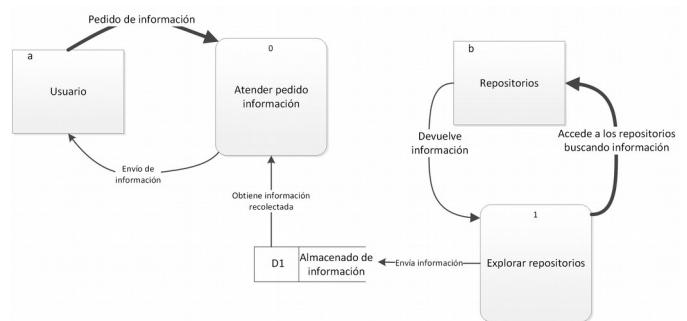


Figura IV.I.1 – Diagrama de flujo de datos

#	Eventos			Flujo de datos		Función asociada
	Tipo	Entidad externa	Descripción	Estímulo	Respuesta	
1	Externo	Usuario	Usuario solicita información	-Pedido información	-Envío información	Atender pedido información
2	Temporal	-	El crawler explora los repositorios	-	-Información explorada	Explorar repositorios

Tabla IV.H.1 – Estímulo-respuesta

J. Diagrama de caso de uso

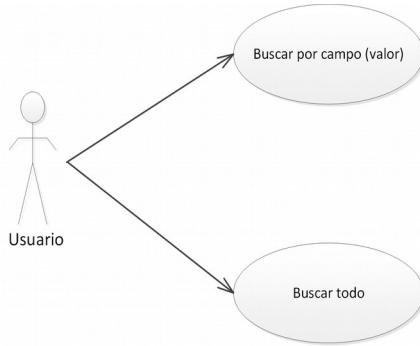


Figura IV.J.1 – Diagrama de caso de uso (verificación de datos).

K. Diagrama de paquete de caso de uso

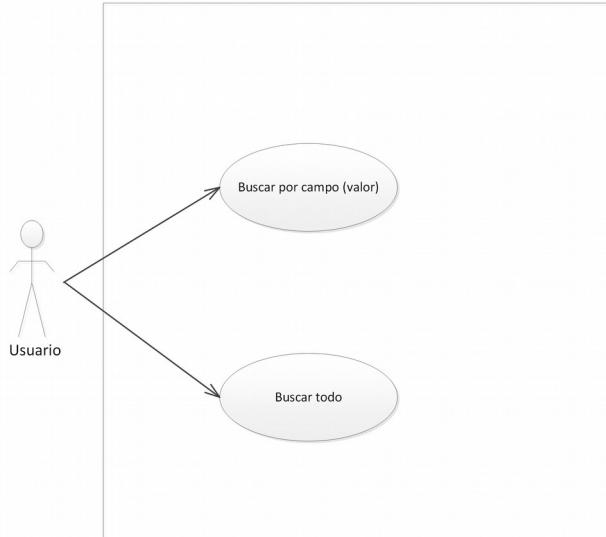


Figura IV.K.1 – Diagrama de paquete de caso de uso.

L. Escenarios de caso de uso

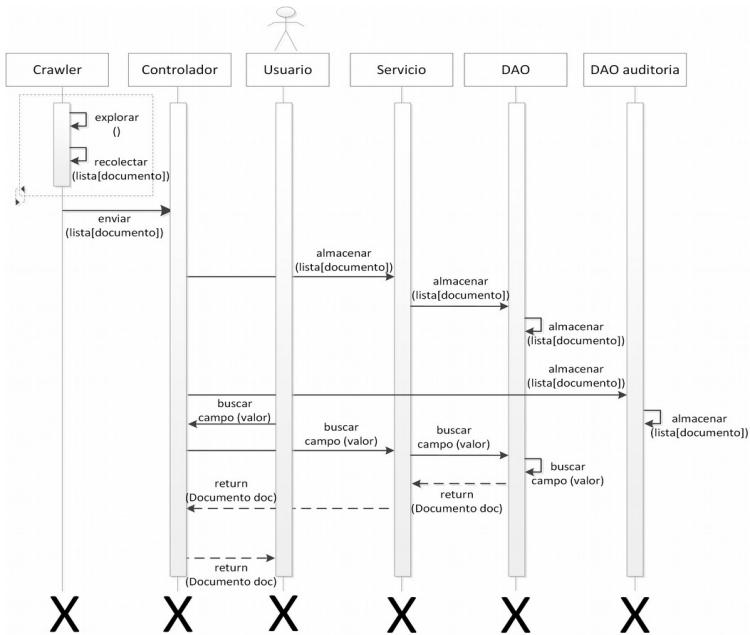
Nombre del caso de uso: Buscar por campo (valor)	ID única: 001
Área: Webbot de datos bibliométricos	
Actor(es): Usuario	
Descripción: El usuario envía al webbot un pedido de información mediante un campo y su valor como parámetro.	
Activar evento: El usuario solicita información al webbot.	
Tipo de señal:	
<input checked="" type="checkbox"/> Externa <input type="checkbox"/> Temporal	
Pasos desempeñados (ruta principal):	
1_El usuario envía un pedido de información al webbot con un campo y su valor como parámetro.	
Precondiciones: El crawler fue lanzado.	
Poscondiciones: El usuario recibe el documento.	
Suposiciones: Los repositorios contienen información. La base de datos contiene información obtenida mediante la exploración del crawler.	
Prioridad: Alta	
Riesgo: Bajo	

Figura IV.L.1 – Escenario de caso de uso buscar por campo (valor).

Nombre del caso de uso: Buscar todo	ID única: 002
Área: Webbot de datos bibliométricos	
Actor(es): Usuario	
Descripción: El usuario envía al webbot un pedido de información de todos los documentos con contenido bibliométrico almacenados.	
Activar evento: El usuario solicita información al webbot.	
Tipo de señal:	
<input checked="" type="checkbox"/> Externa <input type="checkbox"/> Temporal	
Pasos desempeñados (ruta principal):	
1_El usuario envía un pedido de información de todos los documentos al webbot.	
Precondiciones: El crawler fue lanzado.	
Poscondiciones: El sistema usuario recibe los documentos.	
Suposiciones: Los repositorios contienen información. La base de datos contiene información obtenida mediante la exploración del crawler.	
Prioridad: Alta	
Riesgo: Baja	

Figura IV.L.2 – Escenario de caso de uso buscar todo.

M. Diagramas de secuencia



N. Diagrama de clases

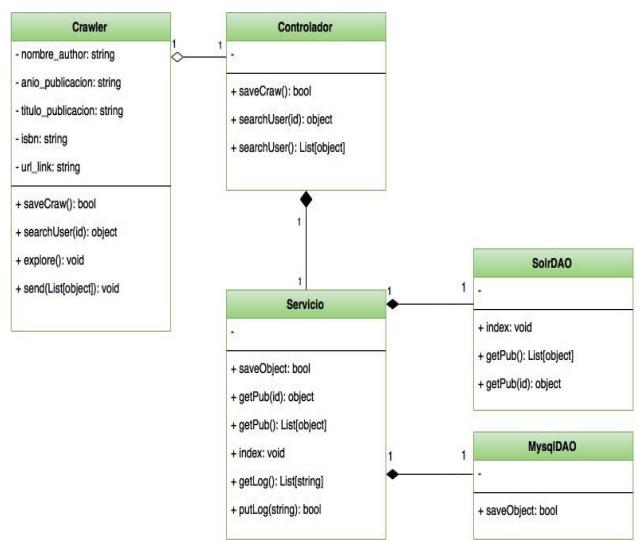
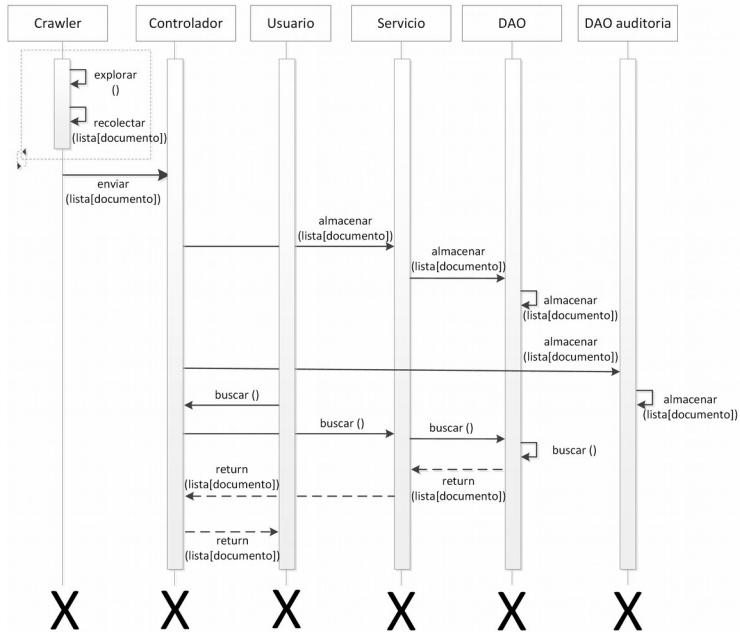


Figura IV.N.1 – Diagrama de clases.

Figura IV.M.1 – Diagrama de secuencia para el caso de uso buscar por campo (valor).



O. Diagrama de entidad-relación

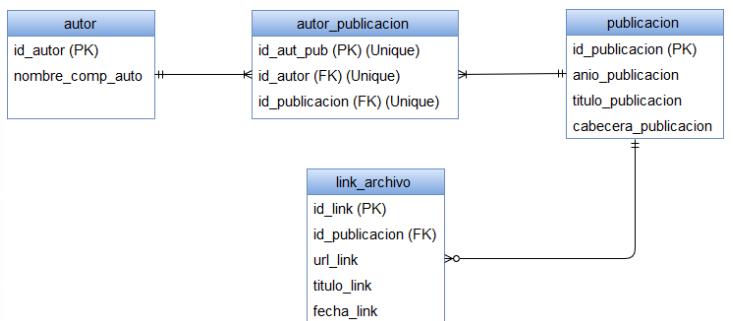


Figura IV.O.1 – Diagrama de entidad-relación.

Figura IV.M.2 – Diagrama de secuencia para el caso de uso buscar todo.

V. Procesos de postdesarrollo

A) Proceso de mantenimiento

Para nuestro proyecto no contemplamos el proceso de mantenimiento porque excede el alcance del proyecto solicitado por el cliente.

B) Proceso de retiro

Para nuestro proyecto no contemplamos el plan de retiro porque excede el alcance del proyecto solicitado por el cliente.

VI. Procesos integrales del proyecto

A) Procesos de verificación y validación

a) Introducción

Para los fines del proyecto interpretamos a la validación como la tarea que se ocupa de controlar si el producto satisface los requerimientos del usuario previamente establecidos, y como verificación a la tarea de controlar que el producto funciona de acuerdo a su especificación inicial.

b) Plan de verificación y validación

Por la naturaleza del proyecto (no posee una interfaz de usuario, sólo una página de pruebas), lo conversado con el cliente y lo investigado posteriormente, concluimos que la mejor forma de verificar y validar nuestro artefacto software es mediante la comprobación de dos casos de prueba (probamos la API directamente y a través de la página de pruebas).

El primer caso de prueba es probar que el webbot traiga todo el contenido recolectado y almacenado en la base de datos.

El segundo caso de prueba, que es un caso particular del anterior, consiste en probar que el webbot traiga todo el contenido recolectado y almacenado en la base de datos asociado a un campo y un valor especificado para el mismo.

c) Plan de pruebas

Aquí nos vamos a dedicar a tratar el plan de pruebas para nuestro artefacto software. Contemplamos los siguientes tipos de pruebas :

Pruebas de unidad, pruebas de integración y prueba del sistema.

1. Prueba de unidad

El desarrollador realizará pruebas individuales de los módulos que componen el sistema.

Se probará que cada módulo funcione correctamente.

2. Prueba de integración

El desarrollador probará de manera integrada los módulos que integran el sistema.

Se probará la correcta comunicación y funcionamiento de los componentes, una vez integrados.

3. Prueba de sistema

Teniendo en cuenta que el webbot de datos biométricos tiene que recolectar datos y enviarlos a un sistema usuario, convenimos con el cliente testear dos casos de prueba :

Un caso de pruebas que traiga toda la información recolectada y otro caso de pruebas que traiga información filtrada por campo y un valor para el mismo.

F420-A 01 19/10/2015	Plan de pruebas del programa. Proyecto: Proyecto Software. Programa: Webbot para datos bibliométricos. Autor: Grupo 5.
Objetivo de la Prueba: Determinar las fallas en el funcionamiento del Webbot.	
Objetos a Probar:	
1. Obtener toda la información almacenada. 2. Obtener toda la información almacenada asociada a un campo y un valor especificado para el mismo.	
Características a probar: El correcto funcionamiento los objetos a probar, en una plataforma GNU/Linux.	
Características a no probar:	
<ul style="list-style-type: none"> • No se tendrán en cuenta otras plataformas que no sea la mencionada. • Velocidad de las operaciones. 	
Cantidad de casos de prueba: Aún no definido.	
Métodos de prueba a utilizar: Adivinación de errores.	
Recursos a utilizar:	
Técnicos:	
<ul style="list-style-type: none"> • C.P.U. : Intel Core I5 2410M. • Memoria R.A.M. : 4 Gb. DDR 3. • Disco rígido : 500 Gb. • Lenguaje de programación: Python. • Java (requerido por Solr). • MySQL (motor de base de datos). • Postman para Chrome (utilizado para hacer peticiones API Rest Post y Get). • Solr (indexador de textos). • Scrapy (librería para crawlers). • Procesador de textos : LibreOffice Writer. • Sistema operativo: Linux Mint 17.2 Cinnamon 64 bits. • Navegador/es: Google Chrome / Mozilla Firefox. 	
Humanos:	
<ul style="list-style-type: none"> • Analistas: Guastadisegni Federico, Rodríguez Nahuel. • Líder de desarrollo: De Domingo Gonzalo. • Encargado de realizar las pruebas: De Domingo Gonzalo. 	
Plan de tiempos:	
<ul style="list-style-type: none"> • Planificación de la prueba : 6hs. • Diseño de las pruebas : 4hs. • Ejecución de las pruebas : 2hs. • Evaluación de las pruebas : 2hs. • Codificación (de ser necesario) : 4hs. 	

TOTAL :	18hs.
Productos a generar durante el proceso de pruebas:	
<ul style="list-style-type: none"> • Plan de pruebas. • Documento de diseño de pruebas. • Especificación de casos de prueba. • Informe de casos de pruebas ejecutados. • Informe de pruebas. 	
Reparto de responsabilidades:	
<ul style="list-style-type: none"> • Planificación de las pruebas: Rodríguez Nahuel. • Diseño de las pruebas: Guastadisegni Federico. • Ejecución de la prueba: De Domingo Gonzalo. • Codificación (de ser necesario): De Domingo Gonzalo. 	
Métodos de prueba a utilizar:	
Se utilizarán los métodos caja blanca y de adivinación de errores para poder determinar las posibles fallas, errores y defectos del sistema relacionado con la funcionalidad.	
Criterios para aprobación de pruebas:	
Los criterios para la aprobación de las pruebas se realizarán de acuerdo al siguiente listado:	
<ul style="list-style-type: none"> • Excelente: Cuando el resultado obtenido es igual al resultado esperado. • Muy bueno: Cuando el resultado obtenido es similar al resultado esperado. • Bueno: Cuando el resultado obtenido no fue el esperado, pero no ha provocado ninguna falla a nivel funcionamiento. • Regular: Cuando el resultado obtenido no fue el esperado, y ha provocado fallo leve en el funcionamiento. • Malo: Cuando el resultado obtenido no fue el esperado, y ha provocado fallo grave en el funcionamiento, como por ejemplo la salida forzada del sistema. 	

Figura VI.C.1 – Plan de pruebas.

F420-C 19/10/2015 Fecha de emisión		ESPECIFICACIÓN DE LOS CASOS DE PRUEBA Proyecto: Proyecto Software. Programa: Webbot para datos bibliométricos. Autor: Grupo 5.		
Item	Objetivo	Acción	Entrada	Resultado Esperado
1	Capturar links de la página.	Recorrer la página y guardar provisoriamente los links.	http://www.sac.org.ar/ http://rinfifi.mdp.edu.ar/	Almacenar provisoriamente todos los tag “”.
2	Extraer ISBN.	Obtener el ISBN de la publicación.	http://www.sac.org.ar/ http://rinfifi.mdp.edu.ar/	ISBN del paper.
3	Atrapar el nombre del autor/autores.	Tomar el nombre del autor de una publicación, de una página.	http://www.sac.org.ar/ http://rinfifi.mdp.edu.ar/	Obtener el nombre del autor/autores.
4	Capturar el título de la publicación.	Diferenciar el nombre de la publicación y almacenarlo.	http://www.sac.org.ar/ http://rinfifi.mdp.edu.ar/	Título de la publicación, completo.
5	Tomar el año de la publicación.	Analizar el código y obtener el año del paper.	http://www.sac.org.ar/ http://rinfifi.mdp.edu.ar/	Obtener el año, en formato AAAA (A = numero entero de 0-9).
6	Relacionar datos.	Relacionar publicaciones, autores y links.	http://www.sac.org.ar/ http://rinfifi.mdp.edu.ar/	Se genera la relación de los datos obtenidos, para luego darle persistencia.
7	Obtener links a archivos.	Adquirir los links donde se encuentran los archivos de las publicaciones.	http://www.sac.org.ar/ http://rinfifi.mdp.edu.ar/	Links que apuntan a los archivos donde se encuentran las publicaciones.
8	Capturar links de la página.	Recorrer la página y guardar provisoriamente los links.	http://www.sac.org.ar/ http://rinfifi.mdp.edu.ar/	Almacenar provisoriamente todos los tag “”.
9	Extraer ISBN.	Obtener el ISBN de la publicación.	http://www.sac.org.ar/ http://rinfifi.mdp.edu.ar/	ISBN del paper.
10	Atrapar el nombre del autor/autores.	Tomar el nombre del autor de una publicación, de una página.	http://www.sac.org.ar/ http://rinfifi.mdp.edu.ar/	Obtener el nombre del autor/autores.
11	Instanciar un objeto con DjangoItem, a través de Scrapy.	Crear un objeto “DjangoItem”, con la información obtenida por el crawler.	Atributos del objeto “publicaciones”.	Objeto “DjangoItem” creado.
12	Guardar un objeto en la BD.	Se almacenará una instancia de “publicaciones” en la BD.	Objeto “Publicación”.	Correcto almacenamiento de un objeto.
13	Obtener toda la información almacenada en la base de datos.	Recorrer la base de datos y traer todo su contenido.	No aplica.	Observar toda la información almacenada.
14	Obtener toda la información almacenada filtrada por id.	Tomar el id, recorrer la base de datos y traer toda la información asociada a ese id.	No aplica.	Observar la información filtrada por id.

Figura VI.C.2 – Especificación de casos de pruebas.

F420-D 19/10/2015 Fecha de emisión		CASOS DE PRUEBA EJECUTADOS Proyecto: Proyecto Software. Programa: Webbot para datos bibliométricos. Autor: Grupo 5.		
Item	Acción	Entrada	Resultado Esperado	Resultado Obtenido
1	Capturar links de la página.	http://www.sac.org.ar/ http://rinfo.fimdp.edu.ar/	Almacenar los links del tipo <a href> y luego recorrerlos.	Se obtienen todos los links de la página.
2	Extraer ISBN.	http://www.sac.org.ar/ http://rinfo.fimdp.edu.ar/	Extraer el ISBN de la página.	Se obtienen todos los dígitos del ISBN.
3	Atrapar el nombre del autor/autores.	http://www.sac.org.ar/ http://rinfo.fimdp.edu.ar/	Recolectar el nombre de el/los autor/es.	Se capturan los nombres correctamente.
4	Capturar el título de la publicación.	http://www.sac.org.ar/ http://rinfo.fimdp.edu.ar/	Obtener el título de la publicación científica.	Captura correcta del título.
5	Tomar el año de la publicación.	http://www.sac.org.ar/ http://rinfo.fimdp.edu.ar/	Tomar correctamente el año de la publicación.	El año es obtenido en el formato AAAA (donde A es un entero de 0 a 9).
6	Relacionar datos.	http://www.sac.org.ar/ http://rinfo.fimdp.edu.ar/	Establecer las relaciones para luego darle persistencia.	Las relaciones se establecen correctamente.
7	Obtener links a archivos.	http://www.sac.org.ar/ http://rinfo.fimdp.edu.ar/	Capturar los links a los archivos donde se encuentran las publicaciones.	Los links se obtienen correctamente.
8	Capturar links de la página.	http://www.sac.org.ar/ http://rinfo.fimdp.edu.ar/	Almacenar los links del tipo <a href> y luego recorrerlos.	Se obtienen todos los links de la página.
9	Extraer ISBN.	http://www.sac.org.ar/ http://rinfo.fimdp.edu.ar/	Extraer el ISBN de la página.	Se obtienen todos los dígitos del ISBN.
10	Atrapar el nombre del autor/autores.	http://www.sac.org.ar/ http://rinfo.fimdp.edu.ar/	Recolectar el nombre de el/los autor/es.	Se capturan los nombres correctamente.
11	Instanciar un objeto con DjangoItem, a través de Scrapy.	Atributos del objeto “publicaciones”.	Objeto “DjangoItem” creado.	Objeto creado satisfactoriamente.
12	Guardar un objeto en la BD.	Objeto “Publicación”.	Correcto almacenamiento de un objeto.	El objeto se almacena correctamente.
13	Obtener toda la información almacenada en la base de datos.	No aplica.	Observar toda la información almacenada.	Se muestra toda la información almacenada.
14	Obtener toda la información almacenada filtrada por id.	No aplica.	Observar la información filtrada por id.	Se muestra la información filtrada por id.

Figura VI.C.3 – Ejecución de casos de pruebas.

F420-E	INFORME DE PRUEBA
19/10/2015	Proyecto: Proyecto Software.
Fecha de emisión	Programa: Webbot para datos bibliométricos.
Autor: Grupo 5.	
<p>Comentarios de la prueba: Los resultados fueron en su mayoría los esperados, si bien en algunos casos no se alcanzó el criterio más alto, el sistema funciona correctamente satisfaciendo todos los ítems que se propusieron al inicio de las pruebas. Mencionamos solamente unas pocas páginas, pero el proceso fue igual para todas las páginas integradas al proyecto.</p> <p>Recomendaciones: No hay recomendaciones.</p>	

Figura VI.C.4 – Conclusión.

Anexo I

Manual de instalación

Webbot para Datos Bibliométricos (WB).

Integrantes :

De Domingo Gonzalo, Guastadisegni Federico, Rodríguez Nahuel.

Cátedra :

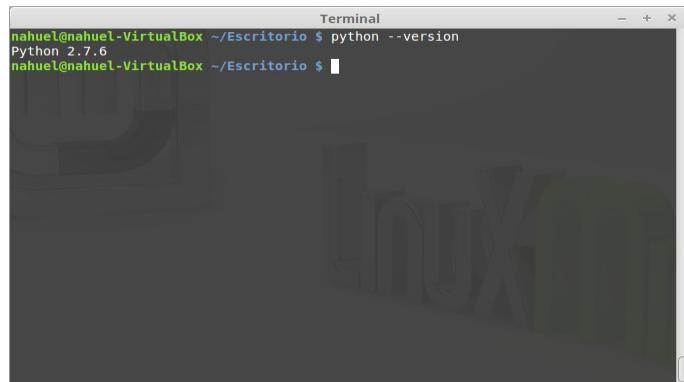
Proyecto de Software
Licenciatura en Sistemas.

2015

Requisitos

Se contemplan como pre-requisitos para la instalación, tener configurado:

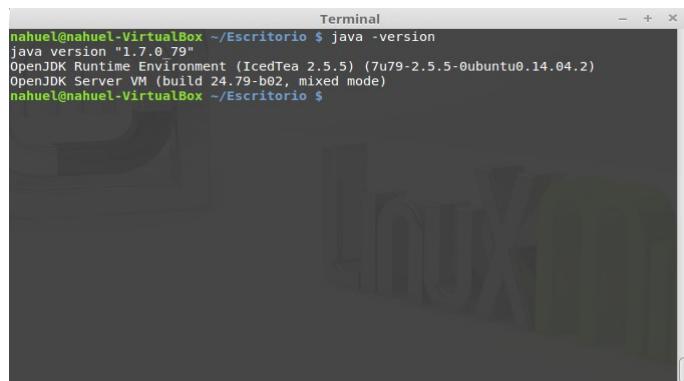
- Python versión 2.7.x.



A screenshot of a terminal window titled "Terminal". The window shows the command "python --version" being run, followed by the output "Python 2.7.6". The background of the window is a blurred image of a Linux logo.

```
Terminal
nahuel@nahuel-VirtualBox ~/Escritorio $ python --version
Python 2.7.6
nahuel@nahuel-VirtualBox ~/Escritorio $
```

- Java versión 1.7.x.



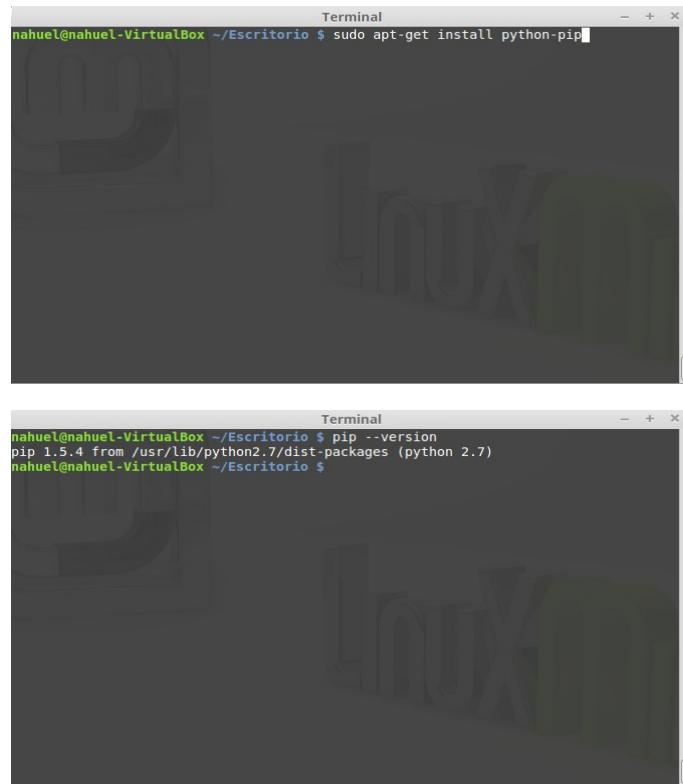
A screenshot of a terminal window titled "Terminal". The window shows the command "java -version" being run, followed by the output "java version "1.7.0_79"" and "OpenJDK Runtime Environment (IcedTea 2.5.5) (7u79-2.5.5-0ubuntu0.14.04.2)". The background of the window is a blurred image of a Linux logo.

```
Terminal
nahuel@nahuel-VirtualBox ~/Escritorio $ java -version
java version "1.7.0_79"
OpenJDK Runtime Environment (IcedTea 2.5.5) (7u79-2.5.5-0ubuntu0.14.04.2)
OpenJDK Server VM (build 24.79-b02, mixed mode)
nahuel@nahuel-VirtualBox ~/Escritorio $
```

Instrucciones

1. Instalar PIP 1.5.4 con el siguiente comando:

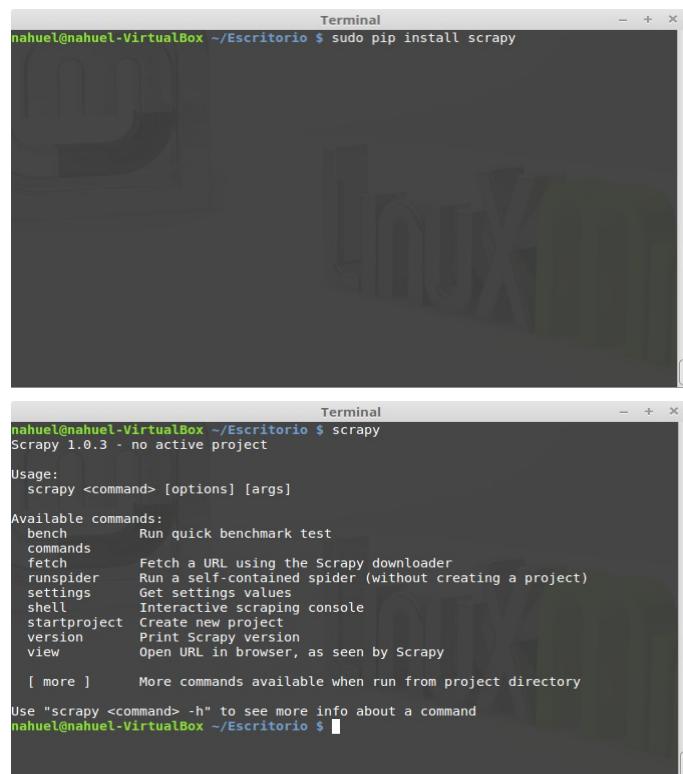
- Sudo apt-get install python-pip.



The image consists of two vertically stacked screenshots of a Linux terminal window. Both screenshots show a dark-themed terminal window with white text. The top screenshot shows the command `sudo apt-get install python-pip` being typed into the terminal. The bottom screenshot shows the output of the command, which includes the message "pip 1.5.4 from /usr/lib/python2.7/dist-packages (python 2.7)".

2. Instalar Scrapy 1.0.3 con el siguiente comando:

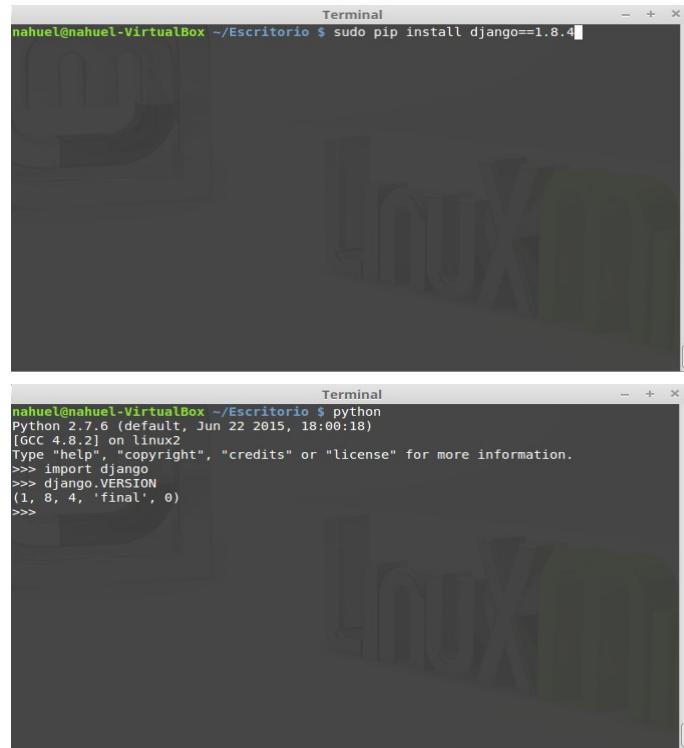
- Sudo pip install scrapy.



The image consists of two vertically stacked screenshots of a Linux terminal window. Both screenshots show a dark-themed terminal window with white text. The top screenshot shows the command `sudo pip install scrapy` being typed into the terminal. The bottom screenshot shows the output of the command, which includes the message "Scrapy 1.0.3 - no active project" and a detailed list of available Scrapy commands such as bench, commands, fetch, runspider, settings, shell, startproject, version, and view.

3. Instalar Django 1.8.4 con el siguiente comando:

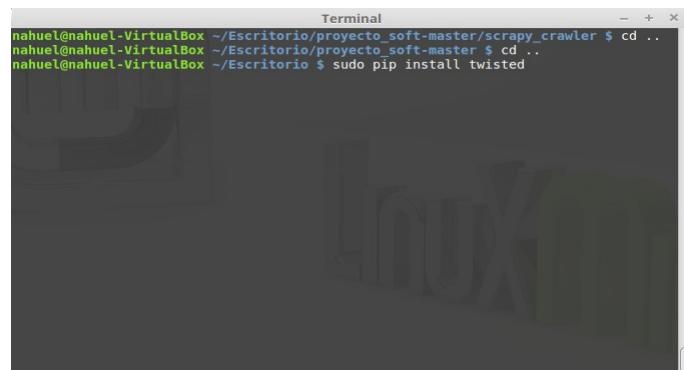
- Sudo pip install Django==1.8.4.



The image shows two terminal windows side-by-side. The top window has the command `nahuel@nahuel-VirtualBox ~/Escritorio $ sudo pip install django==1.8.4` entered. The bottom window shows the Python interpreter running with the command `>>> import django`, followed by the output `>>> django.VERSION` and `(1, 8, 4, 'final', 0)`.

4. Instalar Twisted con el siguiente comando:

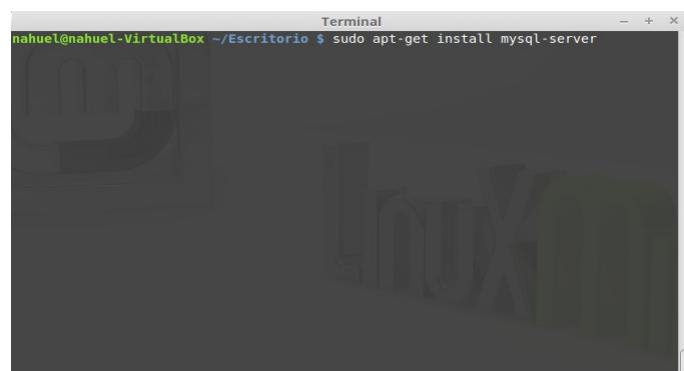
- Sudo pip install twisted.



A single terminal window showing the command `nahuel@nahuel-VirtualBox ~/Escritorio/proyecto_soft-master/scrapy_crawler $ cd ..`, followed by `nahuel@nahuel-VirtualBox ~/Escritorio/proyecto_soft-master $ sudo pip install twisted`.

5. Instalar MySQL con el siguiente comando:

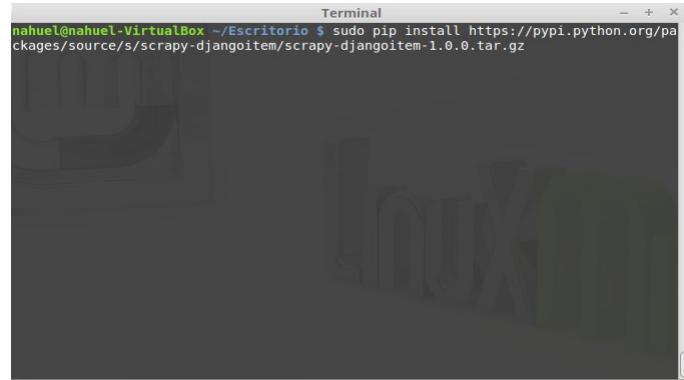
- Sudo apt-get install mysql-server.



A single terminal window showing the command `nahuel@nahuel-VirtualBox ~/Escritorio $ sudo apt-get install mysql-server`.

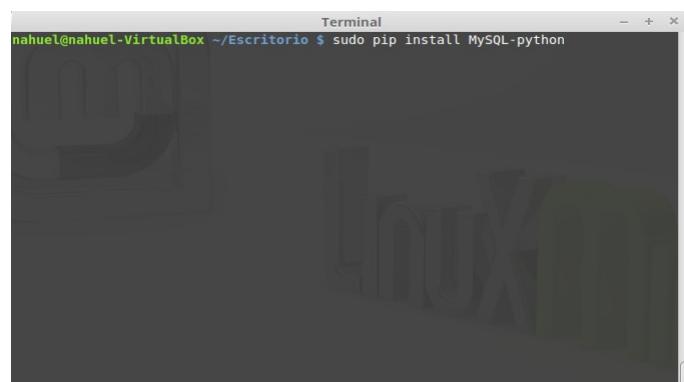
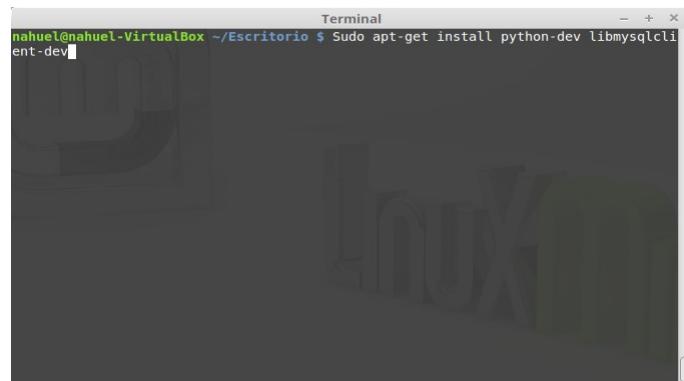
6. Instalar DjangoItem con el siguiente comando:

- Sudo pip install <https://pypi.python.org/packages/source/s/scrapy-djangoitem/scrapy-djangoitem-1.0.0.tar.gz>



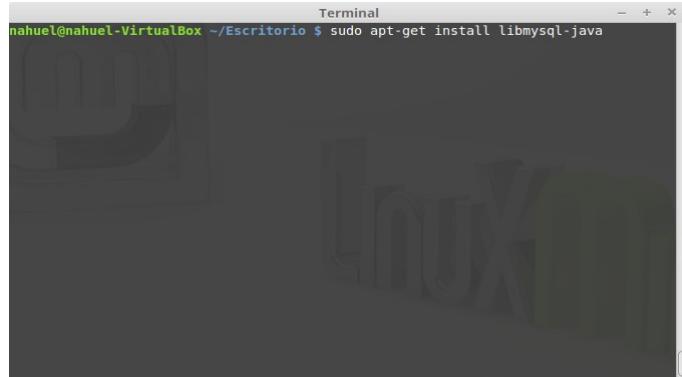
7. Una vez instalado DjangoItem, se requiere un conector para la base de datos, por lo que se deben ejecutar los siguientes comandos:

- Sudo apt-get install python-dev libmysqlclient-dev.
- Sudo pip install MySQL-python.



8. Instalar el conector Java para Solr con el siguiente comando:

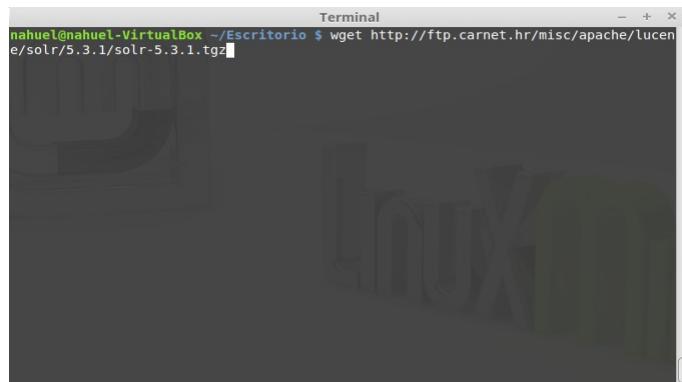
- Sudo apt-get install libmysql-java.



```
Terminal
nahuel@nahuel-VirtualBox ~/Escritorio $ sudo apt-get install libmysql-java
```

9. Descargar el archivo comprimido de Apache Solr con el siguiente comando:

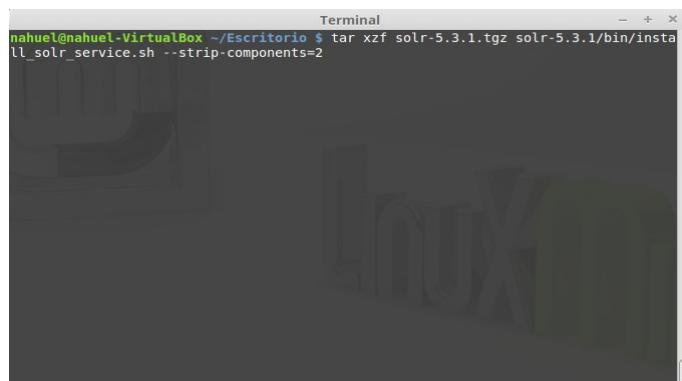
- Wget <http://ftp.carnet.hr/misc/apache/lucene/solr/5.3.1/solr-5.3.1.tgz>



```
Terminal
nahuel@nahuel-VirtualBox ~/Escritorio $ wget http://ftp.carnet.hr/misc/apache/lucene/solr/5.3.1/solr-5.3.1.tgz
```

10. Una vez descargado, hacer un CD hasta el directorio donde esté el paquete, y ejecutar el siguiente comando:

- Tar xzf solr-5.3.1.tgz solr-5.3.1/bin/install_solr_service.sh --strip-components=2.

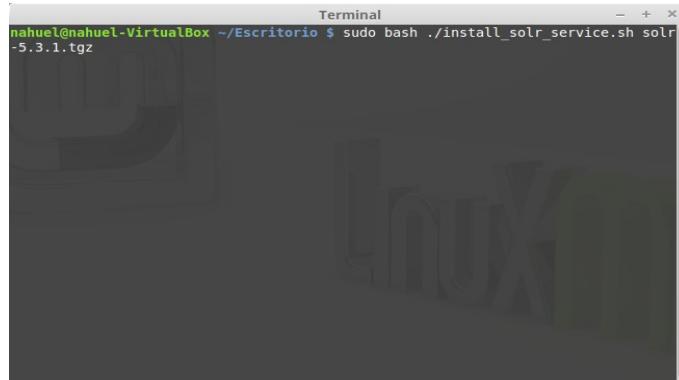


```
Terminal
nahuel@nahuel-VirtualBox ~/Escritorio $ tar xzf solr-5.3.1.tgz solr-5.3.1/bin/install_solr_service.sh --strip-components=2
```

11. Posteriormente, ejecutar el siguiente comando:

- Sudo bash ./install_solr_service.sh solr-5.3.1.tgz.

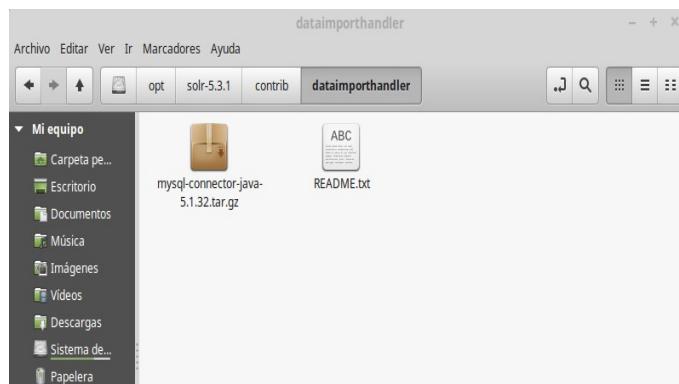
Por defecto, Solr se instalará en /opt/solr y utilizará el puerto 8983.



12. Una vez instalado Solr, bajar el siguiente conector:

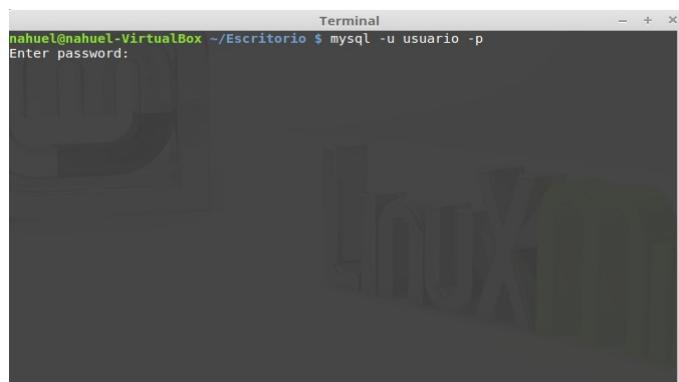
- <http://cdn.mysql.com/archives/mysql-connector-java-5.1/mysql-connector-java-5.1.32.tar.gz>

Ponerlo en /opt/solr-5.3.1/contrib/dataimporthandler (se debe especificar esta ruta en el solrconfig.xml de cada core que se utilice).



13. Ejecutar los siguientes comandos en la consola:

- mysql -u usuario -p (donde usuario es el usuario administrador de las bases de datos). Presionar enter, ingresar contraseña de administrador y presionar nuevamente enter.
- create database WB_app;



```

Terminal
nahuel@nahuel-VirtualBox ~/Escritorio $ mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 38
Server version: 5.5.46-0ubuntu0.14.04.2 (Ubuntu)

Copyright (c) 2000, 2015, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> create database WB_app;

```

14. Ir a /proyecto_soft/Webbot y ejecutar por consola el siguiente comando:

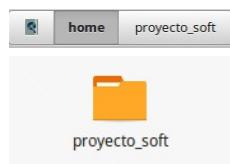
- python manage.py syncdb.

```

Terminal
nahuel@nahuel-VirtualBox ~/Escritorio/proyecto_soft-master/Webbot $ python manag
e.py syncdb

```

15. Para ejecutar el Webbot la carpeta “proyecto_soft” debe encontrarse en el directorio /home, como se muestra a continuación, para el correcto funcionamiento del Script:



Dentro de esta carpeta, encontrará un archivo llamado “Webbot.sh”:



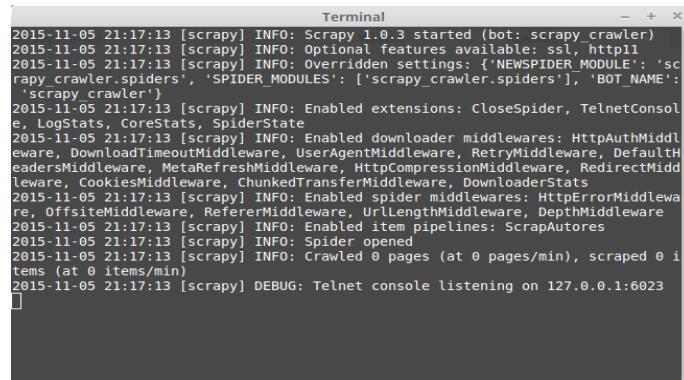
Para poder ejecutarlo, primero hay que abrir la terminal, posicionarnos en la carpeta “proyecto_soft”, y ejecutar el siguiente comando:

- chmod +x Webbot.sh. Este comando hará ejecutable el archivo.

Luego de hacer ejecutable el archivo, simplemente haga doble click sobre el mismo. Se abrirá la siguiente ventana:



Haga click en el botón “Ejecutar en un terminal” y espere a que termine la ejecución del Webbot. Este proceso puede demorar unos minutos.



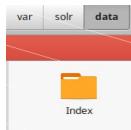
```
Terminal
2015-11-05 21:17:13 [scrapy] INFO: Scrapy 1.0.3 started (bot: scrapy_crawler)
2015-11-05 21:17:13 [scrapy] INFO: Optional features available: ssl, http11
2015-11-05 21:17:13 [scrapy] INFO: Overridden settings: {'NEWSPIDER_MODULE': 'scrapy_crawler.spiders', 'SPIDER_MODULES': ['scrapy_crawler.spiders'], 'BOT_NAME': 'scrapy_crawler'}
2015-11-05 21:17:13 [scrapy] INFO: Enabled extensions: CloseSpider, TelnetConsole, LogStats, CoreStats, SpiderState
2015-11-05 21:17:13 [scrapy] INFO: Enabled downloader middlewares: HttpAuthMiddleware, DownloadTimeoutMiddleware, UserAgentMiddleware, RetryMiddleware, DefaultHeadersMiddleware, MetaRefreshMiddleware, HttpCompressionMiddleware, RedirectMiddleware, CookiesMiddleware, ChunkedTransferMiddleware, DownloaderStats
2015-11-05 21:17:13 [scrapy] INFO: Enabled spider middlewares: HttpErrorMiddleware, OffsiteMiddleware, RefererMiddleware, UrlLengthMiddleware, DepthMiddleware
2015-11-05 21:17:13 [scrapy] INFO: Enabled item pipelines: ScrapAutores
2015-11-05 21:17:13 [scrapy] INFO: Spider opened
2015-11-05 21:17:13 [scrapy] INFO: Crawled 0 pages (at 0 pages/min), scraped 0 items (at 0 items/min)
2015-11-05 21:17:13 [scrapy] DEBUG: Telnet console listening on 127.0.0.1:6023
```

Una vez que el proceso concluya, se cerrará la ventana de la terminal. Eso quiere decir que los datos procesados ya están almacenados en la base de datos.

16. Preparación del core de Solr : En la carpeta “proyecto_soft”, notará que existe una carpeta que se llama Solr. Dentro de la misma, se encuentra la carpeta Index.



Deberá copiar esa carpeta, y pegarla en /var/solr/data:



En el caso de que no se lo permita, por cuestiones de permisos, debe hacer click derecho en un lugar vacío de la carpeta y hacer click en “Abrir con permisos administrativos”. Se le pedirá su contraseña, y se abrirá la carpeta con privilegios elevados:



Lo que debe hacer ahora, es resetear el servicio de Solr, para que este reconozca el core. Para ello, vamos a la terminal y ejecutamos el comando:

- sudo service solr restart.

Esperamos a que termine y el core estará listo para ser utilizado.

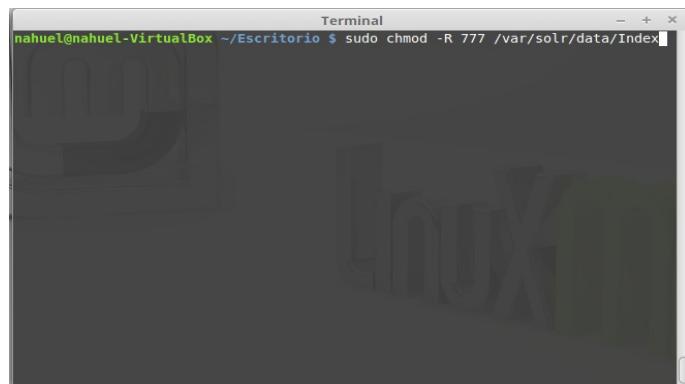
Aclaración I : En el caso de haber creado la base de datos con un usuario/contraseña que no sea root/root, deberá cambiar estos datos en /Index/conf/db-data-config.xml:

```
<dataSource type="JdbcDataSource"
    driver="com.mysql.jdbc.Driver"
    url="jdbc:mysql://localhost:3306/WB_app"
    user="root"
    .....password="root" . />
```

Importante: Para el correcto funcionamiento se deben modificar los permisos del directorio /var/solr/data/Index con el siguiente comando :

- Sudo chmod -R 777 /var/solr/data/Index (Index debe ir con la primera letra en mayúscula, por ser Linux un sistema case sensitive).

Presionar enter, ingresar la contraseña de administrador y confirmar con enter el comando.



Reiniciar el servicio de Solr, para que este reconozca los cambios. En la terminal ejecutar el comando:

- sudo service solr restart.

Anexo II

Manual de usuario

Webbot para Datos Bibliométricos (WB).

Integrantes :

De Domingo Gonzalo, Guastadisegni Federico, Rodríguez Nahuel.

Cátedra :

Proyecto de Software
Licenciatura en Sistemas.

2015

Requisitos

- Conexión a Internet.
- La carpeta “proyecto_soft” tiene que estar en el directorio /home para poder funcionar correctamente.
- El core “Index” que se encuentra dentro de la carpeta “proyecto_soft” debe estar posicionada en /var/solr/data.
- El Webbot ya tiene que haber sido ejecutado, y el core ya tiene que haber sido Indexado para que las URLs funcionen correctamente.
- El localhost provisto por Python deberá estar corriendo (Para ello, vea el apartado “Dominio del Servicio”).

Información general

La API de nuestro Webbot es un servicio REST que proporciona métodos para acceder a recursos específicos en URLs y para la búsqueda y filtrado en base a distintos parámetros. Todas las representaciones se codifican como objetos JSON.

Dominio del servicio

Al no poder utilizar un servidor, utilizaremos el localhost que ofrece Python y Django. Para correr el mismo, primero debe posicionarse en el siguiente PATH:

```
/home/proyecto_soft/Webbot $
```

Una vez allí, debe ejecutar el siguiente comando:

- **python manage.py runserver**.

Luego de ejecutar el comando, la dirección <http://localhost:8000/> quedará lista para utilizarse.

Peticiones

Actualmente, el usuario solamente podrá hacer peticiones HTTP de tipo GET.

Estructura de la respuesta

La mejor manera de entender los resultados devueltos por la API es probar las peticiones que presentaremos mas adelante en este documento. Aún así, ofreceremos una inmersión más profunda en la estructura de resultado API y su organización. Cabe destacar que todas las respuestas tienen formato **JSON**.

La estructura de una respuesta convencional es la siguiente:

Atributo	Tipo	Descripción
Start	Int	Valor desde el cual debe comenzar la petición.
NumFound	Int	Cantidad de resultados esperados.
Docs	Array	Arreglo con todos las publicaciones encontradas (en base al tipo de búsqueda).

Cada publicación tiene la siguiente estructura:

Atributo	Tipo	Descripción
Id	Int	Campo identificadorio de cada una de las publicaciones.
Autor	String	Campo multivaluado que muestra cada uno de los autores de cada publicación.
Título	String	Campo que indica el nombre de la publicación.
Año (Anio, por cuestiones de Encoding)	String	Año en que fue publicado el artículo.
URL	String	Dirección hacia el PDF de la publicación.
ISBN	String	Campo que muestra, dependiendo de la publicación, el ISBN o el ISSN de la misma.
Version	Long	Campo necesario de Solr para su funcionamiento.

Peticiones posibles

A continuación, describiremos todas las peticiones GET posibles de nuestra API, considerando el dominio que señalamos al comienzo de este documento:

Petición: /get/id=#

Tipo: GET.

Notas de implementación: Obtiene una publicación o artículo con el id especificado.

Tipo de respuesta: JSON.

Parámetro: Id=Valor. Debe ser un entero. En caso de no existir, no devolverá nada.

Error/es: Not Found (404), en caso de ingresar de manera incorrecta la URL.

Server error (500), en caso de que la query sea incorrecta (por ejemplo, no ingresar ningún parámetro).

Petición: /get/autor=#

Tipo: GET.

Notas de implementación: Obtiene la/las publicaciones o artículos con el nombre o parte del nombre especificado.

Tipo de respuesta: JSON.

Parámetro: Autor = nombre (puede ser tanto nombre o apellido por separado, como ambos). En caso de no existir, no devolverá nada.

Error/es: Not Found (404), en caso de ingresar de manera incorrecta la URL.

Server error (500), en caso de que la query sea incorrecta (por ejemplo, no ingresar ningún parámetro).

Petición: /get/list

Tipo: GET.

Notas de implementación: Obtiene todas las publicaciones o artículos.

Tipo de respuesta: JSON.

Parámetro: En caso de que el core esté vacío, no devolverá nada.

Error/es: Not Found (404), en caso de ingresar de manera incorrecta la URL.

Server error (500), en caso de que la query sea incorrecta (por ejemplo, no ingresar ningún parámetro).

Petición: /get/anio=#

Tipo: GET.

Notas de implementación: Obtiene la/las publicaciones o artículos con el año especificado.

Tipo de respuesta: JSON.

Parámetro: Anio = valor. En caso de no existir, no devolverá nada.

Error/es: Not Found (404), en caso de ingresar de manera incorrecta la URL.

Server error (500), en caso de que la query sea incorrecta (por ejemplo, no ingresar ningún parámetro).

Petición: /get/url=#

Tipo: GET.

Notas de implementación: Obtiene la/las publicaciones o artículos con la url especificada

Tipo de respuesta: JSON.

Parámetro: Url = dirección. Actualmente, es necesario ingresar completa la url para que funcione. En caso de no existir, no devolverá nada.

Error/es: Not Found (404), en caso de ingresar de manera incorrecta la URL.

Server error (500), en caso de que la query sea incorrecta (por ejemplo, no ingresar ningún parámetro).

Petición: /get/titulo=#

Tipo: GET.

Notas de implementación: Obtiene la/las publicaciones o artículos con el título especificado.

Tipo de respuesta: JSON.

Parámetro: Título = nombre de la publicación. En caso de no existir, no devolverá nada.

Error/es: Not Found (404), en caso de ingresar de manera incorrecta la URL.

Server error (500), en caso de que la query sea incorrecta (por ejemplo, no ingresar ningún parámetro).

Anexo III

Ejecución de pruebas

Webbot para Datos Bibliométricos (WB).

Integrantes :

De Domingo Gonzalo, Guastadisegni Federico, Rodríguez Nahuel.

Cátedra :

Proyecto de Software
Licenciatura en Sistemas.

2015

I. Especificaciones del entorno de pruebas

A) Especificaciones de hardware

- C.P.U. : Intel Core I5 2410M.
- Memoria R.A.M. : 4 Gb. DDR 3.
- Disco rígido : 500 Gb.
- Sistema operativo : Linux Mint 17.2 Cinnamon 64 bits.

B) Especificaciones de software

En nuestro entorno de pruebas vamos a utilizar :

- Java (necesario para el funcionamiento de Solr).
- MySQL (motor de base de datos).
- Librerías para Webbots.
- Solr (indexador de textos).
- Scrapy (librería para crawlers).

II. Preparación del entorno

Para dejar el entorno listo para las pruebas se debe ver el Anexo I que corresponde al manual de instalación.

III. Ejecución de pruebas

A) Pruebas de unidad

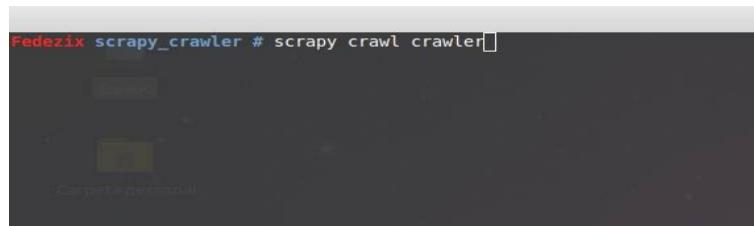
Esta prueba fue realizada de manera informal, utilizando para ello el método de caja blanca, por los programadores a medida que desarrollaron el proyecto. De esta manera, cada uno probó los módulos que desarrolló asegurando de esta manera que cada módulo funciona de la manera que tiene que funcionar.

B) Pruebas de componentes

En esta sección vamos a mostrar las pruebas realizadas a los componentes del sistema.

En las figuras III.B.1, III.B.2, III.B.3, III.B.4, III.B.5, III.B.6 y III.B.7 se pueden ver las pruebas realizadas a Scrapy, la base de datos, una página para pruebas, Solr y Postman.

Con Solr, tuvimos problemas para mapear la base de datos, así como para definir la forma en la que se importan las publicaciones, pero una vez que logramos eso, no tuvimos ningún otro problema.



```
Fedexix scrapy_crawler # scrapy crawl crawler
```

Figura III.B.1 – Scrapy.

```

Terminal
File "/usr/local/lib/python2.7/dist-packages/django/db/backends/utils.py", line 79, in execute
    return super(CursorDebugWrapper, self).execute(sql, params)
File "/usr/local/lib/python2.7/dist-packages/django/db/backends/utils.py", line 64, in execute
    return self.cursor.execute(sql, params)
File "/usr/local/lib/python2.7/dist-packages/django/db/backends/mysql/base.py", line 124, in execute
    return self.cursor.execute(*args)
File "/usr/local/lib/python2.7/dist-packages/MySQLdb/cursors.py", line 205, in execute
    self.errorhandler(self, exc, value)
File "/usr/local/lib/python2.7/dist-packages/MySQLdb/connections.py", line 36, in defaulterrorhandler
    raise errorclass, errorvalue
IntegrityError: (1062, "Duplicate entry 'Eladio Saynchez, Pasnayotis Fandtis, Ibrahim Tarhini, Ijaz Khan' for key 'nombre_comp autor'")
2015-10-22 23:45:50 [scrapy] DEBUG: Closing spider (finished)
2015-10-22 23:45:50 [scrapy] INFO: Dumping Scrapy stats:
{'downloader/request_bytes': 24982,
 'downloader/request_count': 77,
 'downloader/request_method_count/GET': 77,
 'downloader/response_bytes': 913210,
 'downloader/response_count': 77,
 'downloader/response_status_count/200': 74,
 'downloader/response_status_count/301': 3,
 'dupefilter/filtered': 2,
 'finish_reason': 'finished',
 'finish_time': datetime.datetime(2015, 10, 22, 23, 45, 50, 900958),
 'item_scraped_count': 31,
 'log_count/DEBUG': 19,
 'log_count/ERROR': 34,
 'log_count/INFO': 7,
 'offsite/domains': 6,
 'offsite/filtered': 8,
 'redirect_count': 0,
 'response_received_count': 74,
 'scheduler/dequeued': 77,
 'scheduler/dequeued/memory': 77,
 'scheduler/enqueued': 77,
 'scheduler/enqueued/memory': 77,
 'start_time': datetime.datetime(2015, 10, 22, 23, 45, 15, 482259)}
2015-10-22 23:45:50 [scrapy] INFO: Spider closed (finished)

```

Figura III.B.2 – Scrapy.

```

Terminal
<Selector xpath=u'descendant-or-self::a[@href]' data=u'http://www.sac.org.ar/argentine-cardiolo'>
<Selector xpath=u'descendant-or-self::a[@href]' data=u'http://www.sac.org.ar/fca'>
2015-10-22 23:45:18 [scrapy] DEBUG: Filtered offsite request to 'www.fundacioncardiologica.org: <GET http://www.fundacioncardiologica.org/fca>
<Selector xpath=u'descendant-or-self::a[@href]' data=u'http://www.colargcardiol.com.ar'>
2015-10-22 23:45:18 [scrapy] DEBUG: Filtered offsite request to 'www.colargcardiol.com.ar: <GET http://www.colargcardiol.com.ar>
<Selector xpath=u'descendant-or-self::a[@href]' data=u'mailto:info@sac.org.ar'>
2015-10-22 23:45:18 [scrapy] DEBUG: Crawled (200) <GET http://www.sac.org.ar/contact/> (referer: http://www.sac.org.ar/argentine-cardiology-journal-archive)
2015-10-22 23:45:18 [scrapy] DEBUG: Crawled (200) <GET http://www.sac.org.ar/3-congreso-multidisciplinario-de-cardiologia/> (referer: http://www.sac.org.ar/argentine-cardiology-journal-archive)
2015-10-22 23:45:18 [scrapy] DEBUG: Crawled (200) <GET http://www.sac.org.ar/41-congreso-argentino-de-cardiologia/> (referer: http://www.sac.org.ar/argentine-cardiology-journal-archive)
2015-10-22 23:45:18 [scrapy] DEBUG: Crawled (200) <GET http://www.sac.org.ar/biblioteca/> (referer: http://www.sac.org.ar/argentine-cardiology-journal-archive)
2015-10-22 23:45:19 [scrapy] DEBUG: Crawled (200) <GET http://www.sac.org.ar/congresos-de-la-sociedad-argentina-de-cardiologia-2015/> (referer: http://www.sac.org.ar/argentine-cardiology-journal-archive)
2015-10-22 23:45:19 [scrapy] DEBUG: Crawled (200) <GET http://www.sac.org.ar/iii-congreso-argentino-de-ecocardioangiografia-e-imagenes-cardiovasculares/> (referer: http://www.sac.org.ar/argentine-cardiology-journal-archive)
2015-10-22 23:45:19 [scrapy] DEBUG: Crawled (200) <GET http://www.sac.org.ar/investigacion/> (referer: http://www.sac.org.ar/argentine-cardiology-journal-archive)
2015-10-22 23:45:19 [scrapy] DEBUG: Crawled (200) <GET http://www.sac.org.ar/comite-de-biotecnica/> (referer: http://www.sac.org.ar/argentine-cardiology-journal-archive)
2015-10-22 23:45:19 [scrapy] DEBUG: Crawled (200) <GET http://www.sac.org.ar/argenteo-cardiology-journal/?numero=166424> (referer: http://www.sac.org.ar/argenteo-cardiology-journal-archive)
2015-10-22 23:45:19 [scrapy] DEBUG: Crawled (200) <GET http://www.sac.org.ar/agenda/> (referer: http://www.sac.org.ar/argentine-cardiology-journal-archive)
2015-10-22 23:45:19 [scrapy] DEBUG: Crawled (200) <GET http://www.sac.org.ar/docencia/> (referer: http://www.sac.org.ar/argentine-cardiology-journal-archive)
2015-10-22 23:45:19 [scrapy] DEBUG: Crawled (200) <GET http://www.sac.org.ar/distritos/> (referer: http://www.sac.org.ar/argenteo-cardiology-journal-archive)
2015-10-22 23:45:19 [scrapy] DEBUG: Crawled (200) <GET http://www.sac.org.ar/consejos-cientificos/> (referer: http://www.sac.org.ar/argenteo-cardiology-journal-archive)
2015-10-22 23:45:19 [scrapy] DEBUG: Crawled (200) <GET http://www.sac.org.ar/staff/> (referer: http://www.sac.org.ar/argenteo-cardiology-journal-archive)
2015-10-22 23:45:19 [scrapy] DEBUG: Crawled (200) <GET http://www.sac.org.ar/area-de-consensos-y-normas/> (referer: http://www.sac.org.ar/argenteo-cardiology-journal-archive)
2015-10-22 23:45:19 [scrapy] DEBUG: Crawled (200) <GET http://www.sac.org.ar/investigacion/> (referer: http://www.sac.org.ar/argenteo-cardiology-journal-archive)
2015-10-22 23:45:19 [scrapy] DEBUG: Crawled (200) <GET http://www.sac.org.ar/comite-de-biotecnica/> (referer: http://www.sac.org.ar/argenteo-cardiology-journal-archive)
2015-10-22 23:45:19 [scrapy] DEBUG: Crawled (200) <GET http://www.sac.org.ar/argenteo-cardiology-journal/?numero=166424> (referer: http://www.sac.org.ar/argenteo-cardiology-journal-archive)
2015-10-22 23:45:22 [scrapy] DEBUG: Crawled (200) <GET http://www.sac.org.ar/politicas-de-salud/> (referer: http://www.sac.org.ar/argenteo-cardiology-journal-archive)
2015-10-22 23:45:22 [scrapy] DEBUG: Crawled (200) <GET http://www.sac.org.ar/argenteo-cardiology-journal/?numero=167088> (referer: http://www.sac.org.ar/argenteo-cardiology-journal-archive)

```

Figura III.B.3 – Scrapy.

	id_publicacion	anio_publicacion	titulo_publicacion	cabecera_publicacion	isbn
<input type="checkbox"/> Editar	93	2015	¿Tiempos de crisis, tiempos de retorno? Trayectori...	10.17533/udea.espo.n47a13	
<input type="checkbox"/> Editar	146	2015	¿Son racionales las normas sociales?	Null	
<input type="checkbox"/> Editar	272	2015	¿Hacia dónde va la investigación del comportamiento...	Null	
<input type="checkbox"/> Editar	399	2015	¿Cómo podemos tener percepciones más o menos claras...	Null	
<input type="checkbox"/> Editar	1724	2013	"Likely to be eaten by a Grue" - the relevance of ...	Null	

Figura III.B.4 – Captura de la base de datos con phpMyAdmin. A la izquierda, se pueden apreciar las tablas generadas por Django.

De Domingo Gonzalo, Guastadisegni Federico, Rodriguez Nahuel. 2015. Análisis de Sistema. Revista Latinoamericana de Ingeniería de Software, 1(1): 1-3, ISSN 2314-2642. Proyecto Software, Universidad Nacional de Lanús.



Figura III.B.5 – Con la finalidad de probar las respuestas de Solr de manera mas cómoda y visual, decidimos crear una página simple que tome esas consultas y muestre los resultados. Esta página es sólo para PRUEBAS, ya que nuestro software final es una API, y no posee interfaz.

Figura III.B.6 – Prueba directamente en el core de Solr.

Figura III.B.7 – Datos traídos con Postman.

Importante: Si se obtienen nuevos datos con el Webbot, se debe realizar lo siguiente:

- Limpiar el index de Solr con:

http://localhost:8983/solr/Index/update?stream.body=%3Cdelete%3E%3Cquery%3E*:*%3C/query%3E%3C/delete%3E

- Para reindexar la base, se debe usar:

<http://localhost:8983/solr/Index/dataimport?command=full-import>

Nota: Si es la primera vez que se ejecuta el Webbot, esto tiene que hacerse obligatoriamente.

C) Pruebas de sistema

A continuación, vamos a mostrar una captura de nuestra API terminada. En este caso, la petición va a ser sobre todas las publicaciones disponibles. La respuesta siempre va a ser un JSON. La idea es que, posteriormente, cualquier aplicación pueda tomar esta API, y utilizar la información para lo que deseé, lo que excede nuestro proyecto. Para poder obtener mas información acerca de la estructura de la respuesta, así como de las posibles peticiones de nuestra API, consulte el Anexo II manual de usuario.

En la figura III.C.1 se puede ver ejecutado el caso de prueba “traer todo” (se visualizan los artículos indexados hasta el momento, 3.559).

En la figura III.C.2” se puede ver ejecutado el caso de prueba “traer por ID (entero)”.

En la figura III.C.3” se puede ver ejecutado el caso de prueba “traer por año (entero 4 dígitos)”.

En la figura III.C.4” se puede ver ejecutado el caso de prueba “traer por autor (string)”.

En la figura III.C.5” se puede ver ejecutado el caso de prueba “traer por ISBN (string)”.

En la figura III.C.6” se puede ver ejecutado el caso de prueba “traer por título (string)”.

En la figura III.C.7” se puede ver ejecutado el caso de prueba “traer por url (string)”.

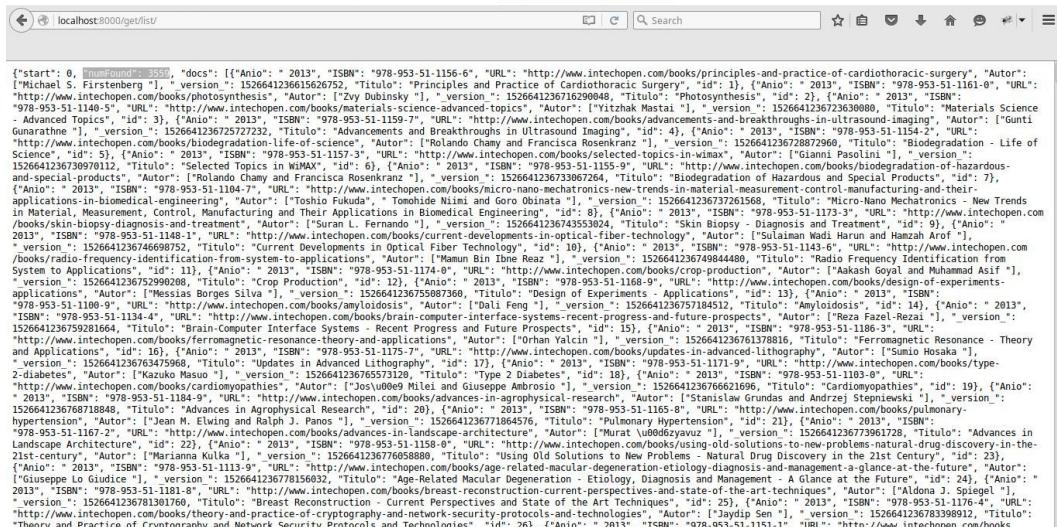


Figura III.C.1 – Caso de pruebas “traer todo” ejecutado.



Figura III.C.2 – Caso de pruebas “traer por ID (1)” ejecutado.

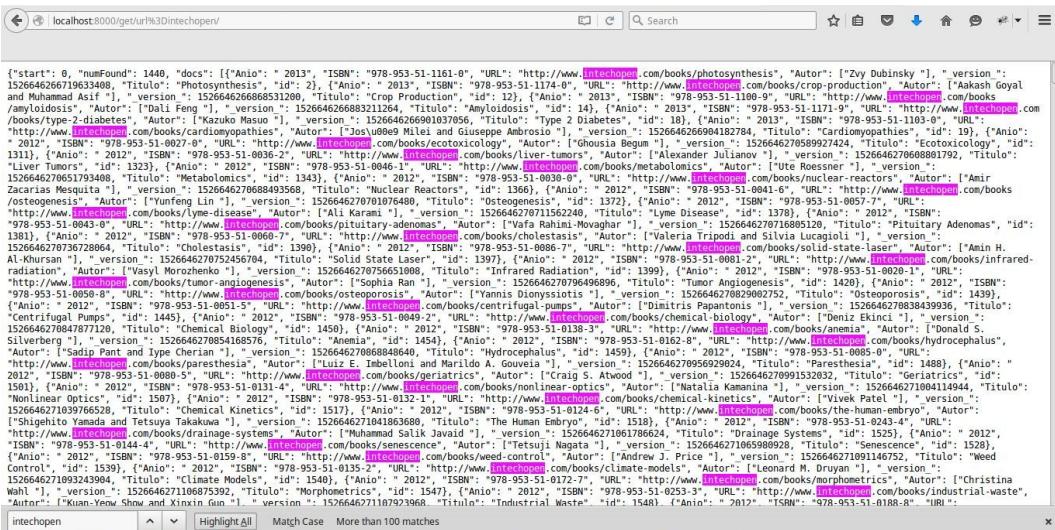


Figura III.C.3 – Caso de pruebas “traer por URL (InTechOpen)” ejecutado.

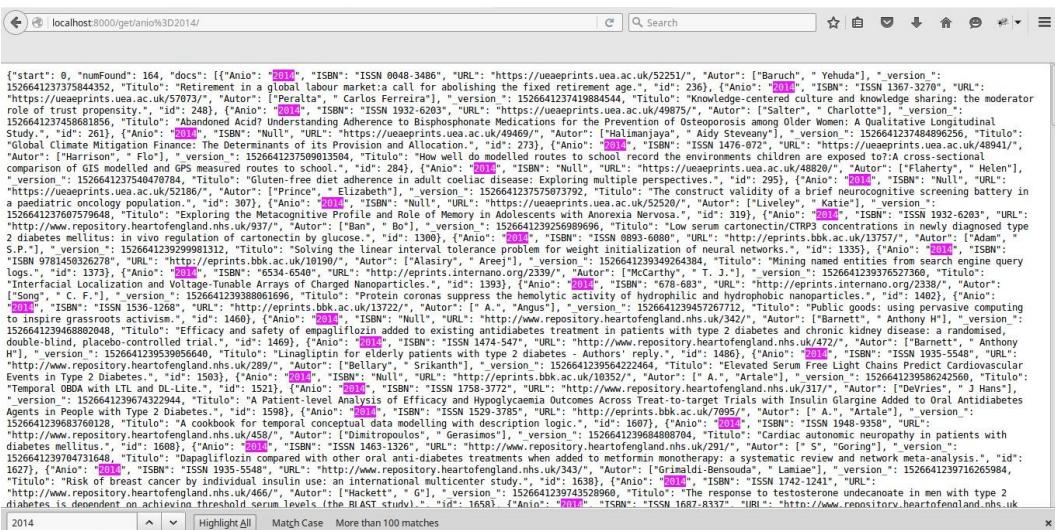


Figura III.C.4 – Caso de pruebas “traer por año (2014)” ejecutado.

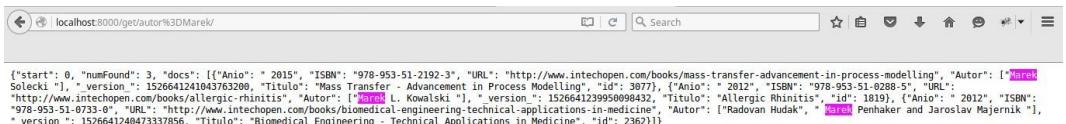


Figura III.C.5 – Caso de pruebas “traer por autor (Marek)” ejecutado.

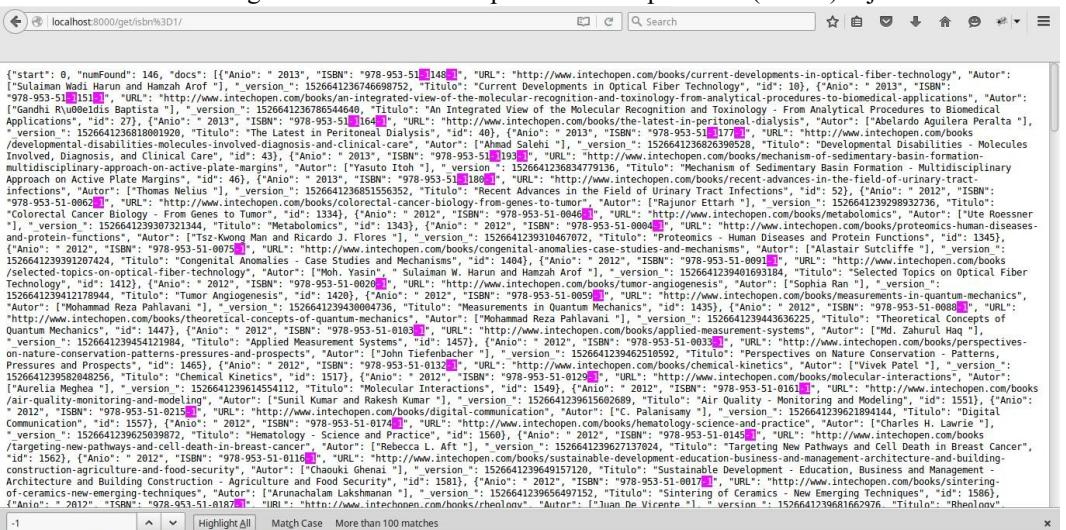


Figura III.C.6 – Caso de pruebas “traer por ISBN (-1)” ejecutado.

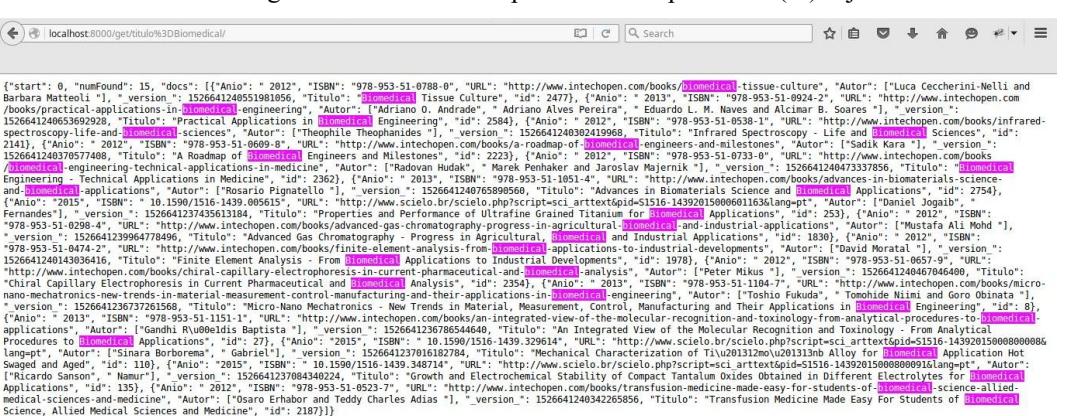


Figura III.C.7 – Caso de pruebas “traer por título (Biomedical)” ejecutado.

De Domingo Gonzalo, Guastadisegni Federico, Rodriguez Nahuel. 2015. *Análisis de Sistema*. Revista Latinoamericana de Ingeniería de Software, I(1): 1-3, ISSN 2314-2642. Proyecto Software, Universidad Nacional de Lanús.

D) Prueba de sistema ejecutada con la página de pruebas

A continuación, vamos a ejecutar los mismos casos de prueba ejecutados en la API mediante nuestra página de pruebas, la cual nos permite mostrar de manera más atractiva la información. Vale recordar que esta página es sólo para PRUEBAS, ya que nuestro software final es una API, y no posee interfaz.

En la figura III.D.1 se puede ver ejecutado el caso de prueba “traer todo” (se visualizan los artículos indexados hasta el momento, 3.559).

En la figura III.D.2” se puede ver ejecutado el caso de prueba “traer por URL”.

En la figura III.D.3” se puede ver ejecutado el caso de prueba “traer por año”.

En la figura III.D.4” se puede ver ejecutado el caso de prueba “traer por autor”.

En la figura III.D.5” se puede ver ejecutado el caso de prueba “traer por ISBN”.

En la figura III.D.6” se puede ver ejecutado el caso de prueba “traer por título”.

The screenshot shows a dark red header with the title 'Web Bot de Datos Bibliométricos' and the logo of 'UNIVERSIDAD NACIONAL DE LANUS'. Below the header is a search bar with placeholder text 'Buscar!' and a magnifying glass icon. The main content area displays search results with the message 'Se encontraron: 3559 resultados'. Below this, a single result is shown with the following details:
ID: 1
Autor: Michael S. Firstenberg
Título: Principles and Practice of Cardiothoracic Surgery
Año: 2013
URL: <http://www.intechopen.com/books/principles-and-practice-of-cardiothoracic-surgery>

Figura III.D.1 – Caso de pruebas “traer todo” ejecutado.

The screenshot shows a dark red header with the title 'Web Bot de Datos Bibliométricos' and the logo of 'UNIVERSIDAD NACIONAL DE LANUS'. Below the header is a search bar with placeholder text 'URL:' and a magnifying glass icon. The main content area displays search results with the message 'Se encontraron: 3539 resultados'. Below this, a single result is shown with the following details:
ID: 2
Autor: Zvy Dubinsky
Título: Photosynthesis
Año: 2013
URL: <http://www.intechopen.com/books/photosynthesis>

Figura III.D.2 – Caso de pruebas “traer por URL” ejecutado.

Web Bot de Datos Bibliométricos

Anio:2016 Buscar!

Se encontraron: 186 resultados

ID: 126

Autor: Geetha, Govindasamy

Título: Investigations on the Synthesis, Optical and Electrical Properties of TiO₂ Thin Films by Chemical Bath Deposition (CBD) method

Año: 2016

URL: http://www.scielo.br/scielo.php?script=sci_arttext&pid=S1516-14392016005002109&lang=pt

Figura III.D.3 – Caso de pruebas “traer por año” ejecutado.

Web Bot de Datos Bibliométricos

Autor:Alencar Buscar!

Se encontraron: 1 resultados

ID: 81

Autor: Wanderson Martins, Alencar

Título: Feeding behavior and productive performance of steers fed pearl millet grain-based diets containing proportions of babassu mesocarp bran

Año: 2015

URL: http://www.scielo.br/scielo.php?script=sci_arttext&pid=S1516-35982015001200425&lang=pt

Figura III.D.4 – Caso de pruebas “traer por autor” ejecutado.

Web Bot de Datos Bibliométricos

ISBN:1 Buscar!

Se encontraron: 146 resultados

ID: 10

Autor: Sulaiman Wadi Harun and Hamzah Arof

Título: Current Developments in Optical Fiber Technology

Año: 2013

URL: <http://www.intechopen.com/books/current-developments-in-optical-fiber-technology>

Figura III.D.5 – Caso de pruebas “traer por ISBN” ejecutado.



Figura III.D.6 – Caso de pruebas “traer por título” ejecutado.

Como se puede ver, se cumplen perfectamente los casos de pruebas generados y acordados con el cliente. El sistema es capaz de devolver artículos por el campo y su valor y también es capaz de devolver todos los artículos indexados.

Por lo tanto concluimos que el sistema cumple con las especificaciones solicitadas por el cliente.

Anexo IV

Repositorios utilizados

Webbot para Datos Bibliométricos (WB).

Integrantes :

De Domingo Gonzalo, Guastadisegni Federico, Rodríguez Nahuel.

Cátedra :

Proyecto de Software
Licenciatura en Sistemas.

2015

Para encontrar repositorios acordes a nuestro proyecto buscamos repositorios académicos y científicos utilizando Google. La mayoría no servía por carecer de campos claves, o bien por tener una organización demasiado precaria imposible de automatizar. Logramos encontrar dos directorios de repositorios realmente completos :

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

OpenDOAR

Directory of Open Access Repositories
[Home](#) | [Find](#) | [Suggest](#) | [Tools](#) | [FAQ](#) | [About](#) | [Contact Us](#)

The Directory of Open Access Repositories - OpenDOAR

[Search for repositories](#) | [Search repository contents](#) | [List of repositories](#) | [Repository Statistics](#)

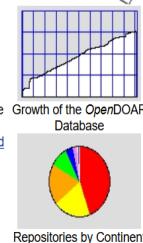
OpenDOAR is an authoritative directory of academic open access repositories. Each OpenDOAR repository has been visited by project staff to check the information that is recorded here. This in-depth approach does not rely on automated analysis and gives a [quality-controlled](#) list of repositories.

OpenDOAR has over 2600 listings!

As well as providing a simple [repository list](#), OpenDOAR lets you [search for repositories](#) or [search repository contents](#). Additionally, we provide tools and support to both repository administrators and service providers in sharing best practice and improving the quality of the repository infrastructure. Further explanation of these features is given in a project document [Beyond the list](#).

The current directory lists repositories and allows breakdown and selection by a variety of criteria - see the [Find](#) page - which can also be viewed as [statistical charts](#). The underlying database has been designed from the ground up to include in-depth information on each repository that can be used for search, analysis, or underpinning services like text-mining. The OpenDOAR service is being developed incrementally, developing the current service as new features are introduced. A list of [Upgrades](#) and [Additions](#) is available.

Developments will be of use both to users wishing to find original research papers and for service providers like search engines or alert services which need easy-to-use tools for developing tailored search services to suit specific user communities.



OpenDOAR is one of the SHERPA Services Including RoMEO and JULIET, run by the [Centre for Research Communications](#) (CRC). Current development work is currently funded by [Jisc](#), with contributions from the CRC host organisation, the [University of Nottingham](#).



OpenDOAR has also been identified as a key resource for the Open Access community ([K.B.Oliver & R.Swain, 2006 - PDF](#)) and identified as the [leader in repository directories](#) in a study by [Johns Hopkins University](#). OpenDOAR was one of the services which contributed to [SHERPA](#) being awarded the [2007 SPARC Europe Award for Outstanding Achievements in Scholarly Communications](#).

More information on the project is available on this site through the [About](#) page.

© 2006-2014, [University of Nottingham, UK](#). Last reviewed: 23-Apr-2014

<http://roar.eprints.org/>

Registry of Open Access Repositories

Home About Search Search Content Browse

Login | New Entry | Create Account

Welcome to the Registry of Open Access Repositories

Notice:

We had a major storage controller failure, which led to the harvesting service failing. We are in the process of recovering data and restarting the harvesting. This failure only affects the tracking of content growth. New repositories should continue to be registered as they are all being fully processed as previously.

We are doing everything we can to bring the service back to normal and we apologise for the interruption to the service.

The aim of ROAR is to promote the development of open access by providing timely information about the growth and status of repositories throughout the world. Open access to research maximises research access and thereby also research impact, making research more productive and effective. [More Information...](#)

Any Country Any Software Any Repository Type Sort by number of n Search

Displaying results 1 to 20 of 4159.
 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | Next

Export 4159 results as Activity table Export RSS 1.0 Atom RSS 2.0

Search Content Graphical analysis

Deposit Activity

1. Networked Digital Library of Theses and Dissertations Union Catalog (1533400 records) - 18 April 2006 [Record Details]

2010 2011 2012 2013 2014 2015

Deposit Activity

Estos directorios contienen repositorios académicos y científicos de todo el mundo, se los puede obtener ordenados por países o por continentes. Hubo que efectuar una búsqueda minuciosa en cada uno porque encontramos que los directorios contienen tanto repositorios adecuados para la indexación como repositorios no aptos para la indexación (de hecho, la mayoría fueron de este tipo).

A continuación indicamos los repositorios que indexamos con nuestro Webbot:

1. Sociedad Argentina de Cardiología:

<http://www.sac.org.ar/argentine-cardiology-journal>

The screenshot shows the homepage of the Sociedad Argentina de Cardiología (SAC) website. At the top, there is a navigation bar with links to SAC, Consejos científicos, Distritos, Docencia, Investigación, Biblioteca, Agenda, Congresos, Consensos, ECAR, and SAC Joven. Below this is a red banner with PROSAC and Revista. The main content area is titled "ARCHIVE" and lists issues of the Argentine Journal of Cardiology from VOLUME 83 (2015) through VOLUME 81 (2013). Each volume entry includes links to Issue 1 through Issue 6. A developer tools window is overlaid on the page, specifically the Element tab of the Inspector panel, highlighting the `<h2 class="tituloVolumen">VOLUME 83 (2015)</h2>` element.

2. Repositorio Institucional INTEMA-Facultad de Ingeniería. Universidad Nacional de Mar del Plata:

<http://rinfo.fimdp.edu.ar/xmlui/recent-submissions>

The screenshot shows the "Recent submissions" page of the RINFO INTEMA-Facultad de Ingeniería UNMdP repository. The page title is "RINFO. INTEMA-Facultad de Ingeniería UNMdP: Recent submissions". It displays a list of 51 recent submissions, each with a title, author(s), and a brief abstract. A developer tools window is overlaid on the page, specifically the Element tab of the Inspector panel, highlighting the `<div class="ds-artifact-item odd">` element.

3. University of East Anglia:

https://ueaepprints.uea.ac.uk/cgi/search/simple?screen=Public%3A%3AEPrintSearch&_action_search=Search&q_merge=ALL&q=fulltext&order=title%2Fcreators_name%2Fdate

4. ROAD - Directory of Open Access Scholarly Resources:

http://road.issn.org/issn_search?afs:query=&show-adv=0&afs:replies=100#.VqaLtl4oDtR

5. InTech – Open science | Open minds:

<http://www.intechopen.com/books/latest/1/list>

The screenshot shows a book listing on the InTechOpen website. The book is titled "Advanced Catalytic Materials - Photocatalysis and Other Current Trends". It is an "Early Online Release" with the editor being Luis Enrique Norena and Jin-An Wang. The ISBN is 978-953-51-2244-9, it has 496 pages, and was published in February 2016. The book is marked as "OPEN ACCESS". The developer tools (Inspector) are open, showing the HTML code for the book listing and the corresponding CSS styles.

6. InterNano – Resources for Nanomanufacturing:

<http://eprints.internano.org>

The screenshot shows the InterNano Nanomanufacturing Library website. The page title is "Welcome to InterNano Nanomanufacturing Library". It features a search bar and a sidebar with links for browsing collections, document types, latest additions, and people. The main content area displays the latest additions to the library, including a list of academic papers. The developer tools (Inspector) are open, showing the HTML structure and CSS styles for the latest additions list.

7. Canada Institute for Scientific and Technical Information:

<http://nparc.cisti-icist.nrc-cnrc.gc.ca/npsi/ctrl>

The screenshot shows the Canada Institute for Scientific and Technical Information's website. The main interface includes a search bar at the top, a sidebar with links for 'Archive', 'Search Tips', 'Linking Tips', 'Frequently Asked Questions', 'Preservation Policy', 'National Science Library Catalogue', 'Order documents', 'Help', 'Digital repository', 'Remote access for NRC', 'Completed Access to Information Requests', and 'Proactive Disclosure'. Below the sidebar, there are two recent publications listed:

1. Photon-momentum transfer in multiphoton ionization and in time-resolved holography with photoelectrons
Chelkowski, Szczepan; Bandrauk, André D.; Corkum, Paul B.
Physical Review A, ISSN: 1050-2947, 92 (5), 2015-11
<http://doi.org/10.1103/PhysRevA.92.051401>
2. Bitumen heavy oil upgrading by cavitation processing: effect on asphaltene separation, rheology, and metal content
Mohapatra, Dipi Prakash; Kimpalani, Deepak M.
Applied Petrochemical Research, ISSN: 2190-5525, 2016-01-28
<http://doi.org/10.1007/s13203-016-0146-1>

At the bottom, there is a 'Find@CISTI' button, a 'View full text' link, and a 'Share It' link. The developer tools panel (F12) is open, showing the DOM structure and the CSS styles applied to the page elements.

8. Birkbeck Institutional Research Online:

<http://eprints.bbk.ac.uk/view/subjects/csis.html>

The screenshot shows the Birkbeck Institutional Research Online website. The main interface includes a search bar at the top, a sidebar with 'Up a level', 'Export as ASCII Citation', 'Export' buttons, and links for 'Atom', 'RSS 1.0', and 'RSS 2.0'. Below the sidebar, there is a navigation menu for 'Birkbeck ePrints Subjects (804)' which includes 'Birkbeck Schools and Research Centres (804)', 'School of Business, Economics & Informatics (804)', and 'Computer Science and Information Systems (804)'. The page also includes a 'Group by: Author(s) | Item Type' link and a 'Jump to: A | B | C | D | E | F | G | H | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z' link. A note indicates 'Number of items at this level: 794.' The developer tools panel (F12) is open, showing the DOM structure and the CSS styles applied to the page elements, specifically focusing on person names.

9. Canterbury Christ Church University:

<http://canterbury33.eprints-hosting.org/view/subjects/QA75.html>

The screenshot shows a library catalog interface. On the left, there's a sidebar with links for 'Browse', 'How to add an item', 'Contact', and 'Login'. The main content area is titled 'Library of Congress Subject Areas (172)' and lists several categories: 'Q Science (172)', 'QA Mathematics (172)', 'QA0075 Electronic computers. Computer science (172)', and 'QA0076.75 Computer software (37)'. Below this is a 'Group by' dropdown menu with options 'Name' and 'Item Type'. A 'Jump to' menu at the bottom includes letters A through Y. The status bar at the bottom indicates 'Number of items at this level: 135.' The bottom part of the screen shows a browser developer tools 'Inspector' panel with the DOM structure and a CSS styles panel.

10. HEFT Repository:

<http://www.repository.heartofengland.nhs.uk/view/subjects/WK.html>

The screenshot shows a library catalog interface. The top navigation bar includes 'Wessex Subject Areas (101)' and 'WK Endocrine system. Endocrinology (101)'. Below this is a 'Group by' dropdown menu with 'Creators' and 'Item Type' selected. A 'Jump to' menu at the bottom includes letters A through W. The status bar at the bottom indicates 'Number of items at this level: 101.' The bottom part of the screen shows a browser developer tools 'Inspector' panel with the DOM structure and a CSS styles panel.

11. BINPAR (Bibliografía Nacional de Publicaciones Periódicas Argentinas Registradas) – CAICYT / CONICET:

<http://binpar.caicyt.gov.ar/cgi-bin/koha/opac-detail.pl?biblionumber=98723>

The screenshot shows the details for the publication 'AFRA, boletín informativo.' by 'Asociación Filatélica de la República Argentina'. The page includes the title, author, type, edition, ISSN, subject, and material type. A screenshot of the browser's developer tools (FIREFOX) is overlaid, showing the DOM structure of the page, specifically focusing on the results summary section.

12. SciELO:

<http://search.scielo.org/q=science&lang=pt&count=50&from=0&output=site&sort=&format=summary&fb=&page=1>

The screenshot shows the SciELO search results for the query 'science'. It displays 55,123 results. The first result is 'Ensaios sobre os nós das redes logísticas' by Orlando Fontes, Lima Jr., published in the Journal of Transport Literature, Dez 2016, Volume 10, Nº 4, Páginas 35 - 39. The page includes a sidebar with filters and collections, and a footer with navigation links.

Anexo V

Versionado de documentos

Webbot para Datos Bibliométricos (WB).

Integrantes :

De Domingo Gonzalo, Guastadisegni Federico, Rodríguez Nahuel.

Cátedra :

Proyecto de Software
Licenciatura en Sistemas.

2015

I. Proceso de selección de modelo de ciclo de vida

Versión	Detalle de la versión		
	Descripción	Autor	Fecha
1.00.001	Primera entrega.	GdeDomingo FGuastadisegni NRodríguez	14/08/15
1.00.002	Se dividió la documentación conjunta en varios documentos individuales correspondientes a los diferentes procesos de la IEEE/1074-1989.	NRodríguez	25/09/15
1.00.003	Se agregó un análisis de los distintos tipos de ciclos de vida disponibles para el desarrollo del proyecto.	NRodríguez	02/10/15
1.00.004	Agregada una figura ilustrativa del ciclo de vida en cascada (figura I.a.1).	NRodríguez	11/10/15

II. Procesos de gestión de proyecto

a) Proceso de iniciación de proyecto

Versión	Detalle de la versión		
	Descripción	Autor	Fecha
1.00.001	Primera entrega.	GdeDomingo FGuastadisegni NRodríguez	14/08/15
1.00.002	Diagrama de Gantt modificado, se agregó más tiempo entre la transición de etapas. El mapa de actividades se reestructuró para adecuarse a nuestro proyecto.	NRodríguez	21/08/15
1.00.004	Se dividió la documentación en documentos individuales correspondientes a los procesos de la IEEE/1074-1989. Se eliminó la sección “Estrategia de Testing”.	NRodríguez	25/09/15
1.00.005	Se modificó la tabla de la figura III.B.4, ahora hay un encabezado con la descripción, la fórmula y el resultado.	NRodríguez	02/10/15
1.00.006	Se eliminó la prueba de componentes de común acuerdo con el cliente.	NRodríguez	24/10/15

b) Procesos de seguimiento y control de proyecto

Versión	Detalle de la versión		
	Descripción	Autor	Fecha
1.00.001	Primera entrega.	GdeDomingo FGuastadisegni NRodríguez	14/08/15
1.00.002	Plan de contingencias modificado : se rompió el formato de la Revista Latinoamerica de Ingeniería de Software para una mejor visualización. Se corrigieron algunos detalles en el encabezado (por ejemplo se reemplazó "%" por "probabilidad").	NRodríguez	25/09/15
1.00.003	Se dividió la documentación conjunta en varios documentos individuales correspondientes a los diferentes procesos de la IEEE/1074-1989. Modificación de la tabla de riesgos y contingencias: El riesgo “Utilización de herramientas no trabajadas, que dificulta la continuidad del trabajo” fue descartado ya que en estos momentos el equipo se encuentra familiarizado con su uso. El riesgo “Cambio del formato de las páginas web” fue modificado a “Cambio del formato de las páginas web, desde las cuales el Webbot obtendrá la información”, a fin de que se refleje de dónde el Webbot adquiere la información. Además, se decidió ponerla en una sección por fuera del formato de la Revista Latinoamericana de Ingeniería de Software.	NRodríguez	02/10/15

c) Proceso de gestión de la calidad del software

Versión	Detalle de la versión		
	Descripción	Autor	Fecha
1.00.001	Primera entrega.	GdeDomingo FGuastadisegni NRodríguez	14/08/15
1.00.002	Se dividió la documentación conjunta en varios documentos individuales correspondientes a los diferentes procesos de la IEEE/1074-1989.	NRodríguez	25/09/15
1.00.003	Se incluyó un párrafo relacionado a la documentación de línea base y al versionado de documentos, para mantener la trazabilidad y un control de documentación eficiente.	NRodríguez	02/10/15

III. Procesos de predesarrollo

a) Proceso de exploración de conceptos

Versión	Detalle de la versión		
	Descripción	Autor	Fecha
1.00.001	Primera entrega.	GdeDomingo FGuastadisegni NRodríguez	14/08/15
1.00.002	Se dividió la documentación conjunta en varios documentos individuales correspondientes a los diferentes procesos de la IEEE/1074-1989.	NRodríguez	25/09/15
1.00.003	Se modificó el contenido en función a lo conversado con el cliente (se incluyó en la sección correspondiente la mención a : página de pruebas y obtención de información rápida, confiable y ordenada) y se hicieron correcciones menores en otros apartados.	NRodríguez	25/09/15
1.00.004	Se modificó la sección “dominio del problema”. Se incluye ahora la cantidad máxima de páginas a ser utilizadas por el Webbot. Además, se separó en una oración el tratamiento que se le hace a la información (recolectar, almacenar e indexar, para luego mostrar) y en otra desde dónde se obtiene dicha información.	NRodríguez	02/10/15

b) Proceso de asignación de sistema

Versión	Detalle de la versión		
	Descripción	Autor	Fecha
1.00.001	Primera entrega.	GdeDomingo FGuastadisegni NRodríguez	14/08/15
1.00.002	Se dividió la documentación conjunta en varios documentos individuales correspondientes a los diferentes procesos de la IEEE/1074-1989.	NRodríguez	25/09/15

IV. Procesos orientados al desarrollo de software

a) Proceso de requisitos

Versión	Detalle de la versión		
	Descripción	Autor	Fecha
1.00.001	Primera entrega.	GdeDomingo FGuastadisegni NRodríguez	14/08/15
1.00.002	Se dividió la documentación conjunta en varios documentos individuales correspondientes a los diferentes procesos de la IEEE/1074-1989.	NRodríguez	25/09/15
1.00.003	El requisito “El usuario no podrá utilizar directamente el Webbot, ni tampoco participará en los procesos internos del artefacto software.” se trasladó desde el apartado “Requisitos no funcionales” al apartado “Requisitos en negativo”.	NRodríguez	02/10/15

b) Proceso de diseño

Versión	Detalle de la versión		
	Descripción	Autor	Fecha
1.00.001	Primera entrega	GdeDomingo FGuastadisegni NRodríguez	14/08/15
1.00.002	Diagramas de caso de uso modificados, ahora se representa los casos de uso de la API de testeo. Eliminados los escenarios de caso de uso que no representan los casos de uso de la API de testeo. Diagramas de secuencia corregidos : se corrigió el tiempo de vida en cada entidad.	NRodríguez	21/08/15
1.00.003	Se dividió la documentación conjunta en varios documentos individuales correspondientes a los diferentes procesos de la IEEE/1074-1989.	NRodríguez	25/09/15

1.00.004	<p>Se modificó la palabra “webbot” por “webbot”, en los dos escenarios de caso de uso.</p> <p>Se modificó el “enviar (documento)” por “return (documento)” en los diagramas de secuencia (sólo cuando representaban una respuesta de una entidad a otra).</p> <p>Modificados los diagramas de secuencia, reordenando las entidades para que la primera acción coincida con el inicio del tiempo de vida en el diagrama. Se modificó el nombre de cada método, poniendo sus nombres en minúscula.</p> <p>Diagramas de clases y de entidad-relación: Modificados los nombres de los atributos y los métodos. Ahora respetan el siguiente formato : NombreAtributo, nombreMétodo().</p>	NRodríguez	02/10/15
1.00.005	<p>Eliminado el diagrama de arquitectura de sistema por común acuerdo con el cliente.</p> <p>Diagrama de clases : eliminados los métodos de get y put log en la clase “MysqlDAO” por haberse modificado por requerimiento al momento de la implementación.</p> <p>DER : la entidad “autor_publicacion” ahora dispone de una clave primaria numérica autogenerada y autoincremental.</p> <p>DFD modificado (D1 ahora es “almacenado de información”).</p>	NRodríguez	09/10/15
1.00.006	DER : Atributos de la tabla autor_publicaciones ahora son únicos (unique).	NRodríguez	25/10/15
1.00.007	Modificados diagramas de caso de uso, de paquete de casos de uso y de secuencia y un escenario de caso de uso. Se cambió donde decía “buscar por ID” por “buscar por campo (valor)”.	NRodríguez	20/02/16

V. Procesos de postdesarrollo

a) Proceso de mantenimiento

Versión	Detalle de la versión		
	Descripción	Autor	Fecha
1.00.001	Primera entrega	GdeDomingo FGuastadisegni NRodríguez	30/10/15

b) Proceso de retiro

Versión	Detalle de la versión		
	Descripción	Autor	Fecha
1.00.001	Primera entrega.	GdeDomingo FGuastadisegni NRodríguez	30/10/15

VI. Procesos integrales de proyecto

a) Procesos de verificación y validación

Versión	Detalle de la versión		
	Descripción	Autor	Fecha
1.00.001	Primera entrega	NRodríguez	10/10/15
1.00.002	Re-elaboración del documento desde cero.	NRodríguez	28/10/15