



ugr | Universidad
de **Granada**

TRABAJO FIN DE MÁSTER
MÁSTER UNIVERSITARIO EN INGENIERÍA INFORMÁTICA

End-to-End Functional Test

Autor

María Victoria Santiago Alcalá

Directores

Fernando Berzal Galiano



Escuela Técnica Superior de Ingenierías Informática
y de Telecomunicación

—
Granada, Julio de 2018

End-to-End Functional Test Coverage and Impact Study

María Victoria Santiago Alcalá

- **Palabras clave:** scenario, regressions, tests, Operf, Callgraph, Clang, Neo4J, C++, Python, Doxygen, Django.

Resumen

Con este proyecto se pretende llevar a cabo el desarrollo de varias herramientas cuyo objetivo es crear un usuario capaz de ejecutar los test de regresión necesarios para probar las nuevas versiones del código del producto principal de la compañía y además crear una herramienta capaz de seleccionar dentro de una suite de 70,000 pruebas, cuales son las pruebas necesarias para probar una determinada parte del código.

Para ello, se ha elaborado una dashboard la cual permite de forma sencilla e intuitiva crear un usuario de regresión con las características necesarias para poder lanzar estas pruebas. También se ha creado una herramienta la cual ha permitido dar a conocer los tests de regresiones de la completa suite que son necesarios para probar las nuevas partes o modificaciones del código del producto.

El resultado que se ha obtenido ha sido satisfactorio puesto que finalmente por un lado con un simple botón se ha podido crear un usuario de regresión rellenando únicamente un formulario con dos campos (Unix Id y máquina desde la que trabaja). El tiempo empleado en este proceso ha resultado reducirse considerablemente puesto que se han automatizado todas las acciones de creación del usuario en las bases de datos, asignación de puertos y demás operaciones en una sola.

Con respecto a la herramienta de cobertura de las pruebas, el resultado también ha sido satisfactorio puesto que ha reducido el tiempo de ejecución, al igual que ha reducido el numero de recursos usado puesto que ha pasado de ejecutarse la suite entera como se venia haciendo para el mas mínimo cambio, a ejecutarse únicamente partes de esta suite donde el mejor caso ha sido en el que únicamente se ha ejecutado un pequeño módulo de esta y donde el peor caso ha venido dado con la ejecución de la suite entera.

End-to-End Functional Test Coverage and Impact Study

María Victoria Santiago Alcalá

Keywords: scenario, regressions, tests, Operf, Callgraph, Clang, Neo4J, C++, Python, Doxygen, Django.

Abstract

The main purpose of this project is the development of several tools whose objective is to create a user capable of execute the regression tests necessary to test the new versions of the company's main product code and also create a tool capable of choosing from a suite of 70,000 tests, which are the tests needed to test a certain part of the code.

A dashboard has been developed which allows, with a simple and intuitive way, to create a regression user with the necessary characteristics to be able to launch these tests. There is another tool created which make possible to know the regression tests of the entire suite that are necessary to test the new parts or modifications of the product code.

The result obtained in general has been satisfactory. On the one hand with a simple button it is possible to create a regression user filling one form with two fields (Unix Id and the machine from which it works). The time spent in this process has been considerably reduced. All the operations have been automated. On the other hand, with the test coverage tool, the execution time and the number of used resources have been reduced. This tool allows the execution of reduced parts of this suite where the best case has been in which only a small module has been executed and where the worst case has been given with the execution of the entire suite.

Yo, María Victoria Santiago Alcalá, alumno del Máster en Ingeniería Informática de la Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación de la Universidad de Granada, con DNI 76656695, autorizo la ubicación de la siguiente copia de mi Trabajo Fin de Máster en la biblioteca del centro para que pueda ser consultada por las personas que lo deseen.



Fdo: María Victoria Santiago Alcalá

Granada a 9 de Julio de 2018.

D. **Fernando Berzal Galiano**, Profesor del Departamento de Ciencias de la Computación de la Universidad de Granada.

Informa:

Que el presente trabajo, titulado **End-to-End Functional Test Coverage and Impact Study**, ha sido realizado bajo su supervisión por **María Victoria Santiago Alcalá**, y autorizamos la defensa de dicho trabajo ante el tribunal que corresponda.

Y para que conste, expide y firma el presente informe en Granada a 9 de Julio de 2018.

El director:

Fernando Berzal Galiano

Agradecimientos

En primer lugar agradecer a mi tutor D. Fernando Berzal Galiano y a mi manager Alexandru Ivanus el apoyo y ayuda que me han ofrecido a lo largo de la realización del proyecto.

Agradecer también el apoyo, la ayuda y los ánimos que me han dado mis padres, mi hermano y mi pareja , día tras día durante el desarrollo de este proyecto.

Gracias a todos.

Índice de contenidos

1. Introducción.....	17
1.1. Breve descripción y especificaciones.....	17
1.2. Entornos.....	18
1.3. Objetivos.....	19
1.4. Beneficios.....	20
1.5. Riesgos y complejidades.....	21
1.6. Entregables.....	21
1.7. Estructura del trabajo.....	22
1.8. Planificación.....	23
2. Estado del arte.....	25
2.1. Descripción del entorno inicial.....	25
2.1.1. Entorno de pruebas de regresión.....	26
2.1.2. Entorno de las dashboards.....	39
3. Análisis de requisitos.....	40
3.1. Propósito.....	40
3.2. Ámbito del Sistema.....	40
3.3. Descripción general.....	40
3.3.1. Perspectiva del productos.....	40
3.3.2. Funciones del productos.....	40
3.3.3. Restricciones.....	43
3.3.4. Suposiciones y dependencias.....	44
3.4. Requisitos Específicos.....	44
3.5. Análisis y diagramas.....	58
3.5.1. Actores.....	58
3.5.2. Diagramas de casos de uso.....	60
3.5.3. Diagramas de actividad.....	63
4. Diseño y estudio de alternativas.....	83
4.1. Diagrama de clases.....	83
4.2. Contratos.....	83
4.3. Estudio de alternativas.....	87
4.4. Tecnología usada.....	92

5. Proceso de desarrollo y resultados.....	95
5.1. Aplicación web para la creación de usuarios de regresión.....	95
5.1.1. Introducción.....	95
5.1.2. Funcionalidades.....	97
5.1.3. Estructura del proyecto.....	103
5.2. Herramienta para pruebas de regresión.....	105
5.2.1. Modulo 0.....	105
5.2.2. Módulo 1.....	115
5.2.3. Módulo 2.....	119
5.2.4. Módulo 3.....	135
6. Manual de usuario.....	138
6.1. Dashboard de nuevos usuarios de regresión.....	138
6.2. End-to-End Functional Test Coverage Tool.....	145
7. Conclusiones y trabajo futuro.....	147
7.1. Conclusiones.....	147
7.2. Trabajo futuro.....	148
Bibliografía.....	149
Glosario de términos.....	151
Anexos.....	152

Índice de ilustraciones

Ilustración 1. Parte del Diagrama de Gantt de la aplicación web.....	20
Ilustración 2. Parte del diagrama de Gantt de la herramienta de tests.....	23
Ilustración 3. Entornos usados.....	23
Ilustración 4. Ejemplo de distribución de versiones del proyecto.....	25
Ilustración 5. Ejemplo de estructura de los directorios.....	26
Ilustración 6. Lista de test de regresión en el subdirectorio cmbmt.....	27
Ilustración 7. Total de escenarios en ngdcsbld/PRD*.....	27
Ilustración 8. Lanzando regresiones, comandos init y comprobacion de gmake.....	29
Ilustración 9. Lanzando regresiones comando gmake.....	30
Ilustración 10. Lanzando regresiones comandos ntt conf y ntt vsy.....	31
Ilustración 11. Lanzando regresiones, comando loadbasedb.....	32
Ilustración 12. Se ejecuta ntt boot.....	33
Ilustración 13. Se ejecuta ntt t lanzando escenarios y obteniendo resultados.....	33
Ilustración 14. Ejecutando Profiler.....	34
Ilustración 15. Proceso finalizado con éxito.....	35
Ilustración 16. Ingreso a la dashboard donde se visualizan las regresiones lanzadas por los usuarios.....	36
Ilustración 17. Ejemplo de ejecuciones ejecutadas por un usuario.....	36
Ilustración 18. Ejemplo de las regresiones lanzadas en la rama DEV.....	37
Ilustración 19. Imagen del panel principal de las dashboards.....	38
Ilustración 20. Herramienta de tests.....	75
Ilustración 21. Consulta información.....	76
Ilustración 22. Crear nuevos usuarios.....	78
Ilustración 23. Modificar usuarios.....	79
Ilustración 24. Eliminar usuario.....	80
Ilustración 25. Migrar usuario.....	81
Ilustración 26. Exportar a fichero csv.....	82
Ilustración 27. Diagrama de clases de la aplicación web.....	83
Ilustración 28. Ejemplo de parsing con Doxygen.....	91
Ilustración 29. Ventana de creación de usuarios.....	96
Ilustración 30. Crear usuario.....	97
Ilustración 31. Vista de listar usuarios.....	98
Ilustración 32. Actualizar información de usuario.....	99
Ilustración 33. Eliminar usuario.....	99
Ilustración 34. Opción de migrar usuarios.....	100
Ilustración 35. Ejemplo de filtro general.....	101
Ilustración 36. Ejemplo de filtro de usuarios por Unix ID.....	101
Ilustración 37. Resultado de la importación de nuevos usuarios.....	102

Ilustración 38. Ejemplo de fichero csv generado.....	103
Ilustración 39. Funcionamiento del módulo 0 donde se generan los ficheros csv.....	105
Ilustración 40. Procesos Operf.....	106
Ilustración 41. Tabla para SYN.....	108
Ilustración 42. Regresiones por PIDS, en COLA y realizadas en HISTORY.....	109
Ilustración 43. Ejemplo de Makefile usado.....	110
Ilustración 44. Ejemplo del los oprofile generados en cada subset de regresiones.....	111
Ilustración 45. Arbol del subset de regresión RGR_FLI.....	112
Ilustración 46. History del set de regresiones.....	113
Ilustración 47. Contenido del subset de regresión y enlace al fichero generado.....	114
Ilustración 48. Ejemplo de fichero de funciones csv.....	115
Ilustración 49. Procedimiento del módulo 1.....	116
Ilustración 50. Código de la herramienta del Módulo 1.....	116
Ilustración 51. Ejemplo del fichero JSON generado.....	117
Ilustración 52. Ejemplo del Módulo 1.....	117
Ilustración 53. Fichero JSON obtenido del módulo 1.....	118
Ilustración 54. Procedimiento seguido por el módulo 2.....	119
Ilustración 55. Error con Clang en ipython.....	120
Ilustración 56. Configuración de la máquina virtual.....	121
Ilustración 57. Añadiendo el proxy.....	122
Ilustración 58. Máquina Docker funcionando.....	123
Ilustración 59. Ejemplo de funcionamiento de Clang extrayendo los tokens.....	124
Ilustración 60. Obtención del nombre de la función senToNGI del fichero FlightAuthorisaion.cpp.....	125
Ilustración 61. Contenido del fichero FlightAuthorisation.cpp.....	126
Ilustración 62. Salida de ipconfig.....	127
Ilustración 63. Fichero de configuración de squid.....	128
Ilustración 64. Construcción de LLVM y Clang.....	129
Ilustración 65. Resize de la memoria.....	130
Ilustración 66. Directorio build.....	132
Ilustración 67. Árbol con ficheros cpp y sus.hpp.....	133
Ilustración 68. Ejemplo del código del fichero parseador.....	133
Ilustración 69. Ejemplo donde se obtienen en el JSON los nombres de las funciones por fichero .cpp.....	134
Ilustración 70. Fichero JSON del módulo 2.....	134
Ilustración 71. Procedimiento del módulo 3.....	135
Ilustración 72. Captura del código correspondiente al módulo 3.....	136
Ilustración 73. Salida de la ejecución del Módulo 3.....	136
Ilustración 74. Procedimiento realizado al ejecutar la herramienta para obtener las regresiones a ejecutar.....	137

Ilustración 75. Ventana de creación de nuevos usuarios de regresión.....	140
Ilustración 76. Vista de la lista de usuarios existentes.....	142
Ilustración 77. Opciones de migración.....	142
Ilustración 78. Ejemplo de filtro general.....	143
Ilustración 79. Ejemplo de filtro por campos.....	143
Ilustración 80. Ejemplo de Actualización/Modificación de datos de un usuario.....	144
Ilustración 81. Ejemplo de Eliminar usuario de regresión.....	145
Ilustración 82. Video.....	146

1. Introducción

Este proyecto fin de máster se ha realizado con el propósito de llevar a cabo el desarrollo de varias herramientas las cuales faciliten la labor de pruebas/tests funcionales de los nuevos módulos de código que se desarrollen para el producto de la compañía Amadeus SL[1].

La necesidad surge debido a que en el entorno en el que se ejecutaban los test de regresión los cuales comprobaban que los módulos de código añadidos o modificados fuesen correctos, tardaba demasiado tiempo por lo que suponía de gran coste lanzar la suite de test de regresión para cada una de estas modificaciones.

El principal objetivo de este proyecto ha sido minimizar el tiempo de ejecución para cada fichero del producto modificado únicamente ejecutando los test funcionales que se hayan involucrados.

Para su realización, se han llevado a cabo varias fases en las cuales se han obtenido los requerimientos básicos, se han analizado las distintas alternativas software y se han desarrollado los prototipos y las herramientas siendo estos una propuesta inicial pero seria, completa y cabal para iniciar el desarrollo del proyecto propiamente dicho, en cumplimiento de la planificación realizada.

A continuación en los siguientes apartados, se va a ir detallando los aspectos relativos al entorno, análisis y desarrollo de proyecto, junto a otras consideraciones.

1.1. Breve descripción y especificaciones

Hoy en día, el conjunto de pruebas funcionales de extremo a extremo en Customers Management Leaves o CML, tiene un volumen de aproximadamente 70,000 pruebas.

Una campaña de regresión completa requiere un total de 60 horas. Por razones prácticas, la ejecución se distribuye a través de una granja de servidores de regresión dedicada formada por 5 servidores de aplicaciones y 3 bases de datos de Oracle EE.

El enfoque de ejecutar el conjunto de regresión completo entre versiones consecutivas a menudo se considera una "exageración", ya que solo partes del código pueden haber cambiado y el impacto puede ser limitado. Además, a pesar de la gran cantidad de pruebas, la cuestión de la cobertura de código que proporciona la prueba en sí sigue sin respuesta. "¿Cuánto del código se está probando en realidad?" y "Cambié este archivo, ¿qué regresiones debo ejecutar?" Son preguntas válidas que no pueden responderse hoy en día.

1.2. Entorno

Amadeus IT Group es un importante proveedor español de TI para la industria global de viajes y turismo.

Esta empresa se estructura en torno a dos áreas: su sistema de distribución global y su área de negocios de soluciones de IT. Como red internacional, Amadeus ofrece servicios de búsqueda, fijación de precios, reserva, emisión de boletos y otros servicios de procesamiento en tiempo real para proveedores de viajes y agencias de viajes a través del área comercial de distribución de Amadeus CRS. A través de su área de negocio de soluciones de IT, también ofrece sistemas de software para empresas de viajes que automatizan procesos tales como reservas, gestión de inventario y control de salidas.

Amadeus tiene sitios centrales en España, Francia, Alemania, India, Boston, Bangkok, Buenos Aires, Dubai, Miami, Estambul, Singapur y Sydney.

Para graduados de negocios y de IT de alto nivel, así como profesionales apasionados, Amadeus ofrece la oportunidad de unirse a numerosos equipos de proyectos para trabajar en soluciones complejas e innovadoras, desde la investigación aplicada hasta el diseño y desarrollo de soluciones hasta la implementación, en un entorno emocionante y de ritmo rápido.

Equipos de investigación y desarrollo, que trabajan esencialmente en tres áreas:

Los equipos de investigación aplicada se centran en las técnicas de optimización y la mejora de la innovación, en el tratamiento de problemas operacionales, el desarrollo de modelos matemáticos o de simulación, el diseño de prototipos.

Los equipos de análisis funcional, lo que llamamos “Definición del producto”, están impulsando el diseño de nuevas soluciones de software y la mejora de las existentes soluciones, en función del análisis de los requisitos del cliente; trabajan en estrecha colaboración con Product Managers y Software Developers.

Como línea de vida del negocio de Amadeus, los equipos de desarrollo de software realmente definen la arquitectura de las soluciones de Amadeus, realizan pruebas técnicas y desarrollan continuamente nuevas herramientas de vanguardia. Las habilidades específicas en torno a las pruebas, el seguro de calidad y la metodología también están ahí para garantizarnos los más altos estándares de calidad.

Amadeus también es el centro de un equipo comercial que impulsa la identificación de nuevas oportunidades comerciales para las TI de aerolíneas y aeropuertos, entre nuevos clientes existentes o potenciales, y el despliegue de estrategias de ventas a nivel regional o mundial.

1.3. Objetivos

Este proyecto está orientado a responder todas las preguntas formuladas en los apartados anteriores. En su forma más simple, lo que pretendemos lograr es un mapeo entre los archivos de regresión (archivos de escenario .scn) y los archivos de códigos (archivos .cpp, .c, .h, .hpp y .i).

Sin embargo, el problema se vuelve más complicado cuando se tienen en cuenta los archivos Makefiles y de configuración (responsables de la generación del código).

Por lo tanto, el mapeo en sí mismo no es una tarea trivial. Si esto se logra dentro del plazo especificado, el próximo paso sería crear un motor de reglas para determinar el subconjunto de pruebas que se ejecutarán como parte de la campaña de regresión nocturna (en otras palabras, reemplazar el conjunto completo de 70k).

De nuevo, esto no es trivial ya que las pruebas a menudo se ejecutan como parte de un grupo y algunas pruebas hacen la configuración necesaria para que se ejecuten las últimas pruebas (estas son fáciles de identificar ya que siguen una convención de nomenclatura). El último paso de este esfuerzo sería crear una herramienta que, con un sistema local y un conjunto de conjuntos de cambios (Mercurial por ahora, Git en el futuro), lanzará una secuencia de regresiones locales para facilitar a los desarrolladores la prueba del impacto de sus cambios.

Una sinergia interesante que debería considerarse, probablemente para un proyecto futuro, como el tiempo probablemente no lo permita, es aplicar esta última herramienta para trabajar en conjunto con las regresiones dockerizadas. El proyecto de dockerización tiene como objetivo convertir las partes integrales del sistema (aplicación, middleware, base de datos, regresiones y herramienta de regresión) en grupos de trabajo autónomos que se distribuirán en una granja de servidores desplegada en nuestra infraestructura o en alguna nube. IaaS (AWS, Google Cloud, Microsoft Azure por nombrar algunos). Los usuarios iniciarían una secuencia de comandos que crearía / reutilizaría las imágenes de la ventana acoplable correctas, crearía un grupo de trabajo, ejecutaría el subconjunto de regresiones e informaría al usuario.

Sin embargo, es poco probable que esto se logre en este proyecto, ya que el proyecto prototípico de dockerización se está elaborando al mismo tiempo. Por lo tanto, faltarán las partes móviles esenciales.

Descritos ya en el anterior apartado los motivos y objetivos que se pretenden alcanzar con este proyecto, a continuación en los siguientes apartados se narran los beneficios esperados, los posibles riesgos y complejidades que se pueden encontrar y finalmente un listado de los productos entregables.

1.4. Beneficios

En caso de que este proyecto sea un éxito, tendrá varios beneficios:

- La duración y el costo de las campañas de regresión nocturna (en términos de recursos) se reducirán drásticamente.
 - Reducción del tiempo de comercialización de nuestro software. El tiempo de respuesta para los cambios se reducirá drásticamente, lo que nos brinda la posibilidad de lanzamientos más frecuentes y, por lo tanto, tiempos de entrega más rápidos.
 - Los recursos se pueden utilizar para otros fines, como valgrind y callgrind, que hoy en día son simplemente imposibles de ejecutar, ya que las campañas de regresión pueden llevar hasta x100 más tiempo que las ejecuciones normales. Una vez más, no todo el conjunto de pruebas deberá ejecutarse bajo estas configuraciones.
- Una comprensión más profunda de nuestra cobertura de código.
 - Los ingenieros de arquitectura, definición de productos y garantía de calidad podrán evaluar la cobertura funcional de nuestras pruebas de regresión, identificar puntos débiles y deficiencias que se fortalecerán.
 - Los desarrolladores podrán evaluar rápidamente el impacto de sus cambios, reduciendo así la probabilidad de un impacto imprevisto.
 - Identificar los escenarios de regresión que cubren las mismas áreas (y posiblemente se puede identificar la misma funcionalidad). Estas pruebas duplicadas / redundantes podrían descartarse si no agregan ningún valor que conduzca a una reducción del tamaño de la campaña de regresión.

1.5. Riesgos y complejidades

Se pueden hallar algunos posibles riesgos con este nuevo enfoque entre los cuales se destacan los siguientes puntos.

- Si el motor de reglas que determina el subconjunto de pruebas que se ejecutará es demasiado "generoso", podríamos terminar ejecutando un porcentaje demasiado grande de pruebas, obteniendo así pequeños beneficios. En este punto, debe tenerse en cuenta que se esperan grandes campañas de regresión cuando se actualicen las bibliotecas centrales del sistema (por ejemplo, versión MWPack, ngd_core, generadores, por nombrar algunas).
- Si, por el contrario, el motor de reglas es demasiado estricto (o no exacto), corremos el riesgo de omitir las regresiones afectadas. Esto es especialmente peligroso ya que podría causar que los fallos no sean detectados. Como estrategia de mitigación, podríamos ejecutar el conjunto de regresión completo cada fin de semana (solo para estar seguros) pero en el contexto de la carga semanal a las regiones enfrentadas con los clientes, este podría no ser un lujo que podamos permitirnos.
- La asignación actualizada entre el código y las regresiones debe actualizarse a menudo, para evitar extensiones del riesgo anterior. Si esta acción es demasiado costosa para realizarla regularmente, esto podría presentar un problema en la viabilidad de este enfoque.

1.6. Entregables

El producto a desarrollar puede ser descompuesto en módulos funcionales independientes. A continuación se nombran cada uno de ellos:

1. Prototipo de “mapeo” entre código y regresiones (versión fija).
2. Versión industrializada del proceso de “mapeo” para adaptarse a futuras versiones.
3. Creación de la versión base del motor de reglas para producir una asignación entre archivos de código y regresiones. Los Makefiles activarán automáticamente una regresión completa.
4. Motor de reglas de refinamiento para hacer frente a los archivos de configuración.
5. Guión para ayudar a que los desarrolladores puedan identificar las regresiones impactadas durante el desarrollo.
6. Aplicación para añadir usuarios (otorgándoles los correspondientes permisos) que ejecuten los test de regresión.

1.7. Estructura del trabajo

A continuación, en esta parte de la documentación, procedemos a resumir brevemente los contenidos que han sido expuestos en el índice.

- Capítulo 1. Introducción. En este capítulo se comienza con una breve introducción general a lo que se va a tratar en el proyecto, su breve descripción y objetivos.
- Capítulo 2. Estado del arte. Tras la introducción y descripción de los objetivos se describen los posibles beneficios, riesgos y complejidades que pueden surjir y los entregables.
- Capítulo 3. Análisis de requisitos. En esta sección se realiza el análisis de las necesidades y servicios requeridos para la propuesta de una solución a los mismos a corto o largo plazo considerando las restricciones existentes en los ámbitos económico, legal y técnico entre otros.
- Capítulo 4. Diseño y estudio de alternativas. En este apartado se hace un despliegue de los requisitos necesarios para llevar a cabo la creación de la herramienta y aplicación. Seguidamente, se realizan los correspondientes diagramas de casos de uso, de paquetes, etc. Resumiendo, en este punto se trata todo el proceso de obtención del software.
- Capítulo 5. Proceso de desarrollo y resultados. En este apartado trataremos todo el proceso de desarrollo de la aplicación que se ha realizado en este proyecto. Haremos hincapié en las herramientas y el proceso de desarrollo que se ha llevado a cabo.
- Capítulo 6. Manual de usuario. Llegados a este punto y tras el correcto desarrollo del producto software final, su evaluación, su rendimiento y la valoración posterior de sus funcionalidades, procedemos a redactar un simple manual de cómo usar la aplicación/producto final.
- Capítulo 7. Conclusiones y trabajo futuro. Finalmente en este capítulo comentaremos cuáles han sido las conclusiones obtenidas, además de exponer las posibles ampliaciones y los trabajos futuros.
- Bibliografía.
- Glosario de términos.
- Anexos.

1.8. Planificación

En este apartado de la memoria se va a describir la planificación que se ha llevado a cabo para el desarrollo del proyecto.

Desde el inicio de este proyecto fijado en el mes de octubre de 2017 hasta su finalización en el mes de junio de 2018, se han ido desarrollando las herramientas divididas en módulos los cuales a su vez se han subdividido en pequeñas funciones. Para el desarrollo de estas partes funcionales se han ido estableciendo períodos de tiempo en los cuales se ha desarrollado cada una de ellas. Tras la realización de cada modulo se han ido teniendo meetings para dar a conocer el status y para acordar las especificaciones y aclarar los contenidos del siguiente modulo.

Se ha de notar también que cada día se ha realizado un “Stand up” en el cual los miembros del equipo (aunque no trabajasen todos en los mismos proyectos, de echo este proyecto solo ha sido llevado a cabo por una persona) han informado del trabajo desarrollado el día anterior, los problemas y los logros obtenidos y ademas se han descrito las tareas (objetivos) a completar en el día actual.

Para llevar un mejor control de la planificación del proyecto, se ha usado un diagrama de Gantt[2] para planificar el tiempo de desarrollo y documentación de la herramienta.

A continuación en las siguientes capturas se muestran algunas partes del mismo, la primera con respecto a la dashboard y la segunda con respecto a la herramienta de ejecución de pruebas funcionales de regresión.

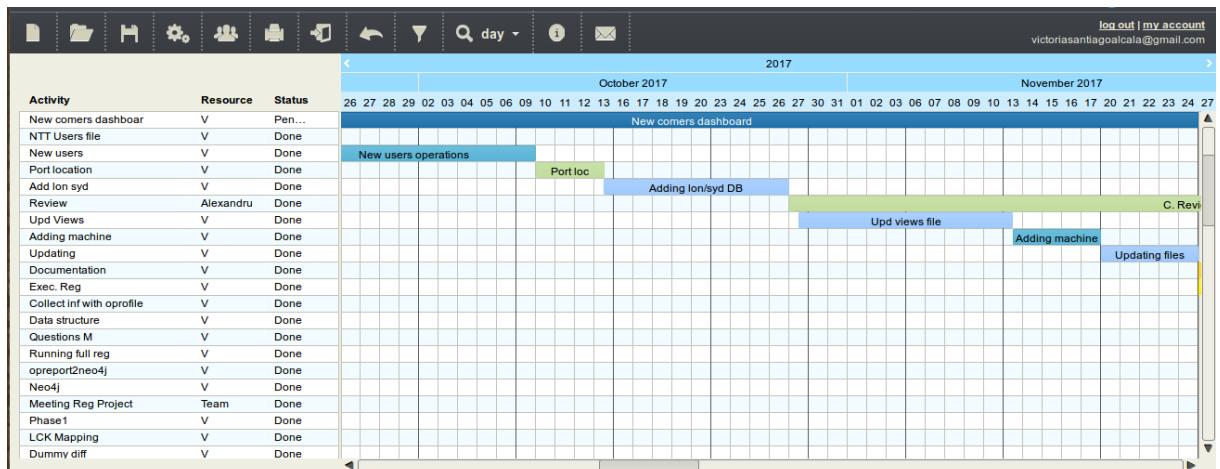


Ilustración 1. Parte del Diagrama de Gantt de la aplicación web

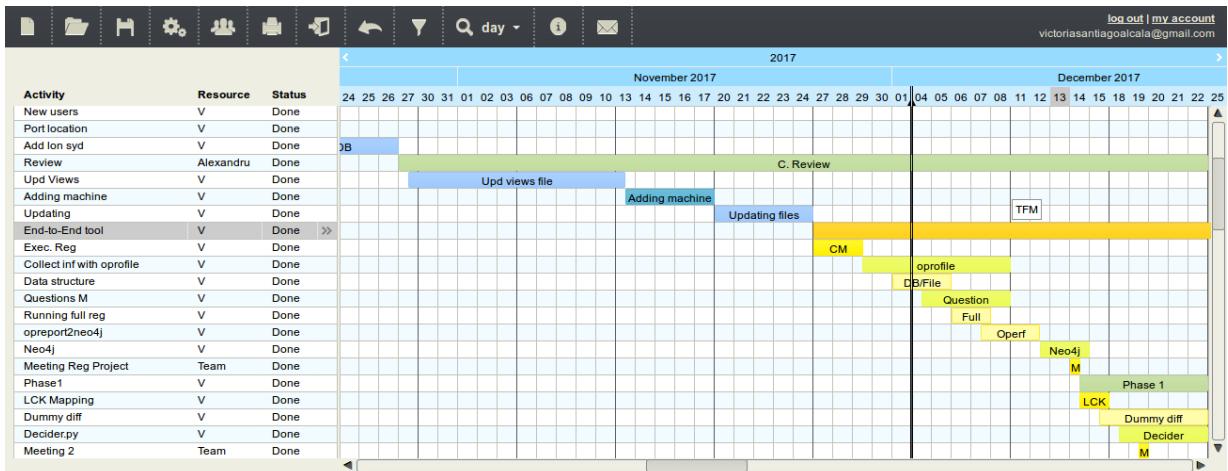


Ilustración 2. Parte del diagrama de Gantt de la herramienta de tests.

También se ha de mencionar que como sistema de control de versiones del código se ha usado Bitbucket y Mercurial mientras que para la documentación oficial se ha usado Confluence (Atlassian Documentation).

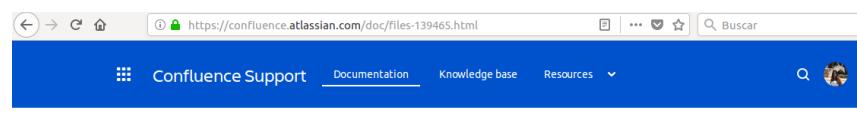
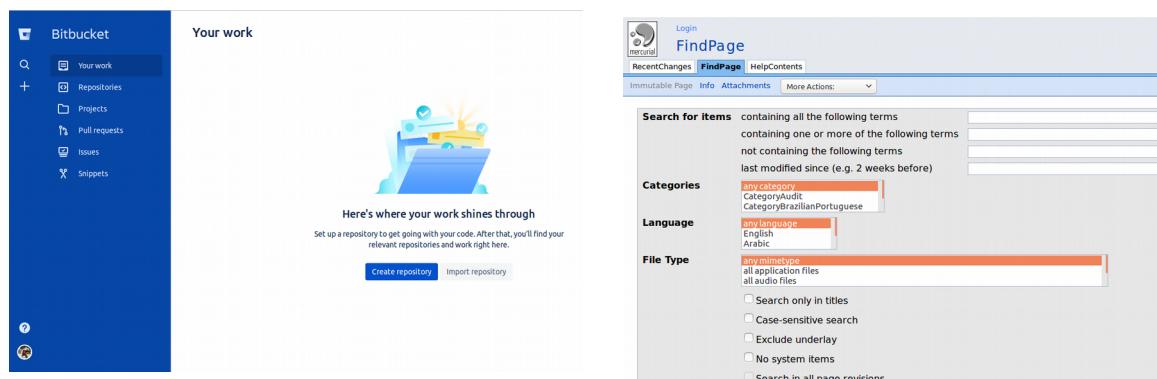


Ilustración 3. Entornos usados.

Como se ha mencionado anteriormente, la compañía no cuenta con un sistema/herramienta el cual permita llevar a cabo los test funcionales de prueba de las partes o módulos nuevos correspondientes al código añadido.

Se cuenta con varias herramientas separadas unas de otras las cuales de forma laboriosa permiten correr los test de regresión.

Inicialmente, se ha de usar la lanzadera de test de regresión la cual construye la suite entera por medio de herramientas como valgrind [3] y callgraph[4], seguidamente el usuario ha de colocar su código en una copia temporal y lanzar la suite entera.

A medida que se vayan ejecutando dichos test uno a uno, se puede consultar en una aplicación web existente el porcentaje de test ejecutados y si estos han tenido o no un resultado satisfactorio.

También se pueden ver los paquetes de regresiones que se van ejecutando en el entorno simplemente yendo al directorio HISTORY para ver los que ya han sido ejecutados o QUEUE para ver los que aun están esperando.

Además, el sistema que permite llevar a cabo el lanzamiento de los test de regresión requiere que el usuario se haya creado en alguna de sus bases de datos por lo que con ello restringe la ejecución a algunos usuarios.

El proceso llevado a cabo para crear un usuario para poder ejecutar los test resulta también laborioso puesto que se ha de crear en un fichero alojado en Bitbucket el usuario que se vaya a crear, introduciendo su Unix ID, la máquina en la que trabaja, los alias que va a tener en las bases de datos, también se tiene que elegir si se crea el usuario en Sydney o en Londres y además se ha de asignar un puerto visualizando previamente los que se hayan ocupados e intentando no sobrecargarlos.

Seguidamente se han de sincronizar los cambios y crearlos también en las respectivas bases de datos con cuidado de insertar los campos de la misma forma que se han insertado en el fichero.

A continuación en el siguiente apartado se va a describir más profundamente los entornos que se tienen.

2.1. Descripción del entorno inicial

En este apartado se van a describir los distintos entornos comenzando por el de las pruebas de regresión y finalizando con el entorno del conjunto de todas las dashboards en el que se va a crear nuestra dashboard de usuarios de regresión.

Se ha de comenzar explicando brevemente como están distribuidos sus directorios.

Cada proyecto de test de regresión se configura por una versión marcada por el año, las correspondientes branch con sus releases.

Dentro de cada uno de ellos a su vez, contienen los distintos módulos correspondientes a los proyectos que desarrolla cada departamento. Un ejemplo puede ser el módulo en el que se desarrolla únicamente la parte del producto relacionada con ticketing, reserva de un asiento en un vuelo o con baggage entre otros.

También se ha de comentar que los módulos a su vez están distribuidos por las distintas máquinas en las que trabajan los departamentos por lo cual no se encuentran unidos hasta que entran en los test, en el entorno de desarrollo y en producción.

A continuación en la siguiente imagen se puede apreciar las distribuciones que se tienen.

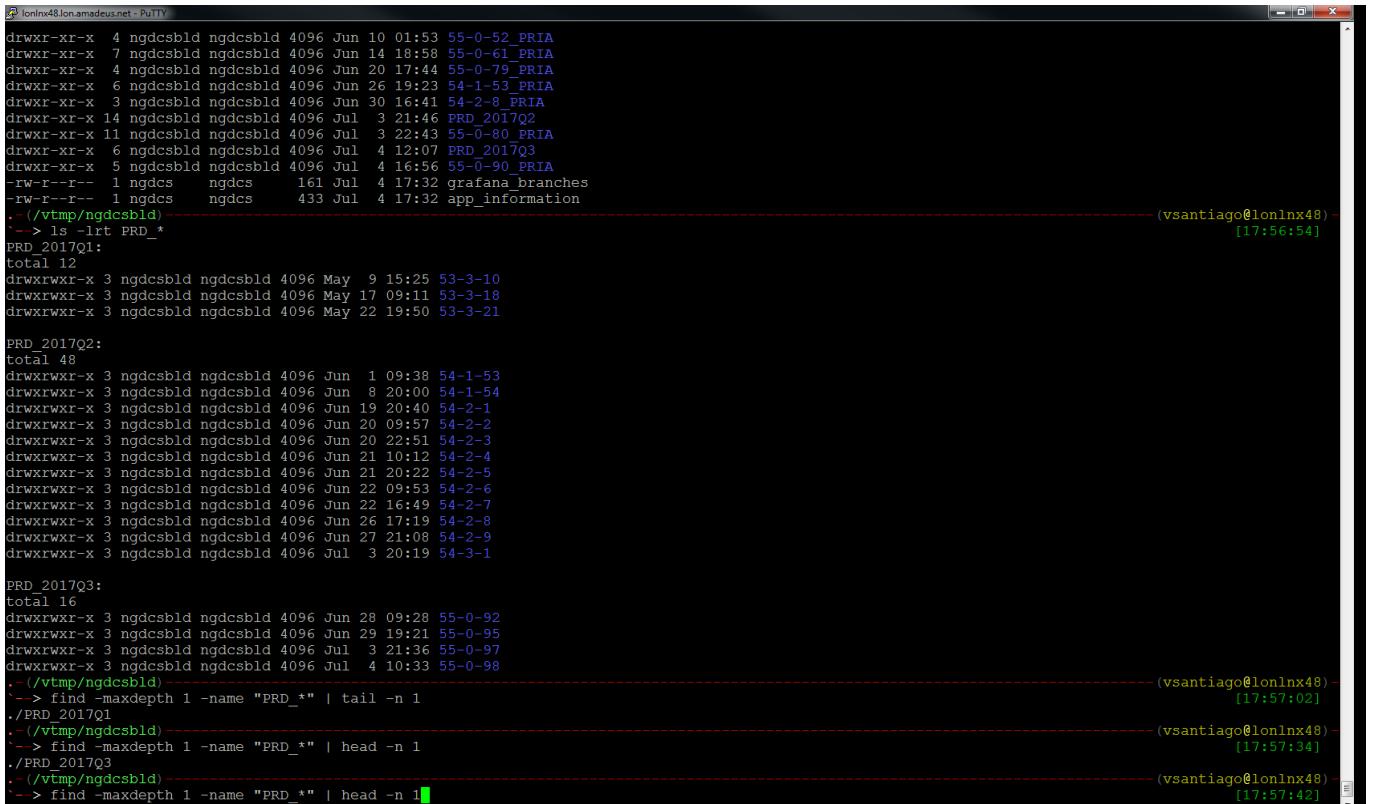
```
[--> cd /vttmp/ngdcsbld/
- ./vttmp/ngdcsbld ----- [17:56:52]
`--> ls -lrt
total 60
drwxr-xr-x 14 ngdcsbld ngdcsbld 4096 Mar 17 12:21 ods_delivery
drwxr-xr-x 5 ngdcsbld ngdcsbld 4096 May 23 02:59 PRD_2017Q1
drwxr-xr-x 3 ngdcsbld ngdcsbld 4096 May 27 02:41 53-3-16_PRI
drwxr-xr-x 8 ngdcsbld ngdcsbld 4096 Jun 5 18:12 integration
drwxr-xr-x 4 ngdcsbld ngdcsbld 4096 Jun 10 01:53 55-0-52_PRI
drwxr-xr-x 7 ngdcsbld ngdcsbld 4096 Jun 14 18:58 55-0-61_PRI
drwxr-xr-x 4 ngdcsbld ngdcsbld 4096 Jun 20 17:44 55-0-79_PRI
drwxr-xr-x 6 ngdcsbld ngdcsbld 4096 Jun 26 19:23 54-1-53_PRI
drwxr-xr-x 3 ngdcsbld ngdcsbld 4096 Jun 30 16:41 54-2-8_PRI
drwxr-xr-x 14 ngdcsbld ngdcsbld 4096 Jul 3 21:46 PRD_2017Q2
drwxr-xr-x 11 ngdcsbld ngdcsbld 4096 Jul 3 22:43 55-0-80_PRI
drwxr-xr-x 6 ngdcsbld ngdcsbld 4096 Jul 4 12:07 PRD_2017Q3
drwxr-xr-x 5 ngdcsbld ngdcsbld 4096 Jul 4 16:56 55-0-90_PRI
-rw-r--r-- 1 ngdcs ngdcs 161 Jul 4 17:32 grafana_branches
-rw-r--r-- 1 ngdcs ngdcs 433 Jul 4 17:32 app_information
. ./vttmp/ngdcsbld ----- [17:56:54]
`--> ls -lrt PRD_*
PRD_2017Q1:
total 12
drwxrwxr-x 3 ngdcsbld ngdcsbld 4096 May 9 15:25 53-3-10
drwxrwxr-x 3 ngdcsbld ngdcsbld 4096 May 17 09:11 53-3-18
drwxrwxr-x 3 ngdcsbld ngdcsbld 4096 May 22 19:50 53-3-21

PRD_2017Q2:
total 48
drwxrwxr-x 3 ngdcsbld ngdcsbld 4096 Jun 1 09:38 54-1-53
drwxrwxr-x 3 ngdcsbld ngdcsbld 4096 Jun 8 20:00 54-1-54
drwxrwxr-x 3 ngdcsbld ngdcsbld 4096 Jun 19 20:40 54-2-1
```

Ilustración 4. Ejemplo de distribución de versiones del proyecto.

Como se puede contemplar en la imagen, tenemos dentro de el sistema de control de salidas el cual esta en la carpeta “ngdcsbld”, varias versiones de producción identificadas por las siglas PRD_201XQX, las cuales vienen finalmente configuradas por los resultados testeados que proporcionan a su vez los subproyectos con las siglas por ejemplo “55-0-52_PRI”.

siempre por defecto se ejecuta el último enlazando este con el directorio default.



```
loninx48.lion.amadeus.net - PuTTY
drwxr-xr-x  4 ngdcbsld ngdcbsld 4096 Jun 10 01:53 55-0-52_PRIA
drwxr-xr-x  7 ngdcbsld ngdcbsld 4096 Jun 14 18:58 55-0-61_PRIA
drwxr-xr-x  4 ngdcbsld ngdcbsld 4096 Jun 20 17:44 55-0-79_PRIA
drwxr-xr-x  6 ngdcbsld ngdcbsld 4096 Jun 26 19:23 54-1-53_PRIA
drwxr-xr-x  3 ngdcbsld ngdcbsld 4096 Jun 30 16:41 54-2-8_PRIA
drwxr-xr-x 14 ngdcbsld ngdcbsld 4096 Jul  3 21:46 PRD_2017Q2
drwxr-xr-x 11 ngdcbsld ngdcbsld 4096 Jul  3 22:43 55-0-80_PRIA
drwxr-xr-x  6 ngdcbsld ngdcbsld 4096 Jul  4 12:07 PRD_2017Q3
drwxr-xr-x  5 ngdcbsld ngdcbsld 4096 Jul  4 16:56 55-0-90_PRIA
-rw-r--r--  1 ngdcn  ngdcn   161 Jul  4 17:32 grafana_branches
-rw-r--r--  1 ngdcn  ngdcn   433 Jul  4 17:32 app_information
...(vtmp/ngdcbsld)----- (vsantiago@loninx48) [17:56:54]
`-> ls -lrt PRD_*
PRD_2017Q1:
total 12
drwxrwxr-x 3 ngdcbsld ngdcbsld 4096 May  9 15:25 53-3-10
drwxrwxr-x 3 ngdcbsld ngdcbsld 4096 May 17 09:11 53-3-18
drwxrwxr-x 3 ngdcbsld ngdcbsld 4096 May 22 19:50 53-3-21

PRD_2017Q2:
total 48
drwxrwxr-x 3 ngdcbsld ngdcbsld 4096 Jun  1 09:38 54-1-53
drwxrwxr-x 3 ngdcbsld ngdcbsld 4096 Jun  8 20:00 54-1-54
drwxrwxr-x 3 ngdcbsld ngdcbsld 4096 Jun 19 20:40 54-2-1
drwxrwxr-x 3 ngdcbsld ngdcbsld 4096 Jun 20 09:57 54-2-2
drwxrwxr-x 3 ngdcbsld ngdcbsld 4096 Jun 20 22:51 54-2-3
drwxrwxr-x 3 ngdcbsld ngdcbsld 4096 Jun 21 10:12 54-2-4
drwxrwxr-x 3 ngdcbsld ngdcbsld 4096 Jun 21 20:22 54-2-5
drwxrwxr-x 3 ngdcbsld ngdcbsld 4096 Jun 22 09:53 54-2-6
drwxrwxr-x 3 ngdcbsld ngdcbsld 4096 Jun 22 16:49 54-2-7
drwxrwxr-x 3 ngdcbsld ngdcbsld 4096 Jun 26 17:19 54-2-8
drwxrwxr-x 3 ngdcbsld ngdcbsld 4096 Jun 27 21:08 54-2-9
drwxrwxr-x 3 ngdcbsld ngdcbsld 4096 Jul  3 20:19 54-3-1

PRD_2017Q3:
total 16
drwxrwxr-x 3 ngdcbsld ngdcbsld 4096 Jun 28 09:28 55-0-92
drwxrwxr-x 3 ngdcbsld ngdcbsld 4096 Jun 29 19:21 55-0-95
drwxrwxr-x 3 ngdcbsld ngdcbsld 4096 Jul  3 21:36 55-0-97
drwxrwxr-x 3 ngdcbsld ngdcbsld 4096 Jul  4 10:33 55-0-98
...(vtmp/ngdcbsld)----- (vsantiago@loninx48) [17:57:02]
`-> find -maxdepth 1 -name "PRD_*" | tail -n 1
./PRD_2017Q1
...(vtmp/ngdcbsld)----- (vsantiago@loninx48) [17:57:34]
`-> find -maxdepth 1 -name "PRD_*" | head -n 1
./PRD_2017Q3
...(vtmp/ngdcbsld)----- (vsantiago@loninx48) [17:57:42]
```

Ilustración 5. Ejemplo de estructura de los directorios.

En la imagen se ha podido ver que la estructura es la misma en los distintos repositorios los cuales se mantienen en las distintas versiones.

Dentro de cada release, tenemos los directorios correspondientes a cada departamento. Se ha de mencionar que este proyecto está destinado en principio al departamento de Customer Management (CM) por lo que se va a trabajar inicialmente en un pequeño módulo de este localizado en el directorio cmapp el cual contiene los escenarios de este gran set de test de regresión.

En la siguiente captura de pantalla se puede ver un ejemplo de los escenarios que se hayan contenidos en uno de los subdirectorios.

```

loninx48.lon.amadeus.net - PUTTY
cmbmt/ngbmtrgr/db/BIDBRQ_112/IdentifyBaggage_SpecialAttribute_ByLicensePlate.Case_001.scn
cmbmt/ngbmtrgr/db/BIDBRQ_112/IdentifyBaggage_LastSeenInformation.Case_005.scn
cmbmt/ngbmtrgr/db/BIDBRQ_112/IdentifyBaggage_ActivationStatus.Case_002.scn
cmbmt/ngbmtrgr/db/BIDBRQ_112/IdentifyBaggage_FlightStatus.Case_002.scn
cmbmt/ngbmtrgr/db/BIDBRQ_112/IdentifyBaggage_EmergencyLock.Err_004.scn
cmbmt/ngbmtrgr/db/BIDBRQ_112/IdentifyBaggage_SpecialAttribute_BycarrierAndTag.Case_001.scn
cmbmt/ngbmtrgr/db/BIDBRQ_112/IdentifyBaggage_PoolIndicator.Case_001.scn
cmbmt/ngbmtrgr/db/BIDBRQ_112/IdentifyBaggage_Missing_Linked_TagNumber.Case_001.scn
cmbmt/ngbmtrgr/db/BIDBRQ_112/IdentifyBaggage_SpecialAttribute_ByUBI.Case_001.scn
cmbmt/ngbmtrgr/db/BIDBRQ_112/IdentifyBaggage_Screening.Case_003.scn
cmbmt/ngbmtrgr/db/BIDBRQ_112/IdentifyBaggage_InternalAirlineText.Case_001.scn
cmbmt/ngbmtrgr/db/BIDBRQ_112/IdentifyBaggage_Screening.Case_005.scn
cmbmt/ngbmtrgr/db/BIDBRQ_112/IdentifyBaggage_Screening.Case_006.scn
cmbmt/ngbmtrgr/db/BIDBRQ_112/IdentifyBaggage_LoadDetails.Case_002.scn
cmbmt/ngbmtrgr/db/BIDBRQ_112/IdentifyBaggage_LoadDetails.Case_008.scn
cmbmt/ngbmtrgr/db/BIDBRQ_112/IdentifyBaggage_UCI.Case_001.scn
cmbmt/ngbmtrgr/db/BIDBRQ_112/IdentifyBaggage_LastSeenInformation.Case_002.scn
cmbmt/ngbmtrgr/db/BIDBRQ_112/IdentifyBaggage_PBT.Case_001.scn
cmbmt/ngbmtrgr/db/BIDBRQ_112/IdentifyBaggage_MOPName_historicalDays.Case_001.scn
cmbmt/ngbmtrgr/db/BIDBRQ_112/IdentifyBaggage_Screening.Case_002.scn
cmbmt/ngbmtrgr/db/BIDBRQ_112/IdentifyBaggage_LastSeenInformation.Case_001.scn
cmbmt/ngbmtrgr/db/BIDBRQ_112/IdentifyBaggage_UBI.Case_006.scn^C

./vtmp/ngdcsbld/PRD_2017Q3/55-0-98/internal)----- (vsantiago@loninx48)
[17:53:22]
68985

```

Ilustración 6. Lista de test de regresión en el subdirectorio cmbmt

Como se puede apreciar en la imagen, se muestran los tests de regresión del directorio cmbmt los cuales corresponden al sistema de baggage.

Se puede a su vez observar también en la última linea de comandos ejecutada que se tienen en los proyectos de Customer Management un total de 68985 posibles escenarios de regresión.

```

In [113]: commands.getoutput('find /vttmp/ngdcsbld/PRD* -maxdepth 1 -name "PRD*" | head -n 1 | tail -n 1 | xargs -I {} find {} -name "*scn" | wc -l')
Out[113]: '207200'

In [114]: 

```

Ilustración 7. Total de escenarios en ngdcsbld/PRD*

Y un total de 207200 en lo que viene a ser los directorios de PRD en general.

Mostrado ya el entorno en el que se encuentran nuestros test de regresión, se va a describir como se venían ejecutando estos test de regresión de forma individual para obtener los resultados de test.

Lanzando regresiones de forma general

Inicialmente se han de ejecutar los siguientes comandos para crearnos nuestro propio directorio y exportar las variables de entorno necesarias:

```
mkvmdir <dir_name> # en mi caso lo llamé cm17q2  
export NG_PROJECT=ngdcs  
  
export  
PYTHONPATH=/data/ngddeldev/int/hgtools/default:/data/ngddeldev/int/pythontools/de-  
fault:/opt/  
devsup/common/latest:/opt/devsup/bms/latest:/opt/devsup/cmk/latest:/opt/devsup/plati-  
num:/da  
ta/ngddeldev/int/ngd_pythonpack/2-0-0-2/lib/python2.7/site-packages
```

Seguidamente se ha de iniciar el directorio de nuestros escenarios en cmapp ejecutando y lanzando el makefile como sigue a continuación:

```
mm init  
mm co cmapp/ngcmtrgr #this is the checkout  
gmake # make -s -j20  
cd cmapp/ngcmtrgr/edi
```

A continuación se han de realizar las configuraciones y el setup de las regresiones para lanzarlas y generar el perfil con oprofile[5].

```
ntt vsy  
ntt conf  
ntt loadbasedb  
ntt boot  
ntt t  
ntt vsy  
ntt PROFILER=oprofile config
```

Si todo ha ido bien obtendremos un resultado como el que se muestra en la siguientes imágenes:

```

lonlnx48.lon.amadeus.net - PuTTY
lrwxrwxrwx 1 alexandru.ivanus ama-unix      62 Nov 14 11:22 acl11321321 -> /vtmp/shadow/lonlnx48/
remote/users/alexandru.ivanus/acl11321321
lrwxrwxrwx 1 alexandru.ivanus ama-unix      66 Nov 14 12:00 57-0-47_PRI_1 -> /vtmp/shadow/lonln
x48/remote/users/alexandru.ivanus/57-0-47_PRI_1
-rw-r--r-- 1 alexandru.ivanus ama-unix      547795 Nov 15 14:36 rdcp.stats.ipython.txt
lrwxrwxrwx 1 alexandru.ivanus ama-unix      58 Nov 16 11:50 q4_otf -> /vtmp/shadow/lonlnx48/remote/
users/alexandru.ivanus/q4_otf
drwxr-xr-x 3 alexandru.ivanus dev_asl      21504 Nov 16 12:02 diffs
lrwxrwxrwx 1 alexandru.ivanus ama-unix      55 Nov 16 12:19 acl -> /vtmp/shadow/lonlnx48/remote/
users/alexandru.ivanus/acl
-rw-r--r-- 1 alexandru.ivanus ama-unix      144073 Nov 16 13:17 aaaa.txt
-rw-r--r-- 1 alexandru.ivanus ama-unix      1827722 Nov 20 10:40 rdcvars.text
lrwxrwxrwx 1 alexandru.ivanus ama-unix      60 Nov 20 14:07 axacsad -> /vtmp/shadow/lonlnxv01/re
mote/users/alexandru.ivanus/axacsad
-rw-r--r-- 1 alexandru.ivanus ama-unix      14425 Nov 20 18:14 edi.txt
lrwxrwxrwx 1 alexandru.ivanus ama-unix      59 Nov 23 16:02 cm17aq1 -> /vtmp/shadow/lonlnx48/remote/
users/alexandru.ivanus/cm17aq1
-rw-r--r-- 1 alexandru.ivanus ama-unix      4805 Nov 23 16:29 ai_env
alexandru.ivanus@lonlnx48:~> mkvdircd cm17q2                                         4:45PM
alexandru.ivanus@lonlnx48:~/cm17q2> mm init -r latest_release                         4:45PM
++ Initialise using release 58-0-15
alexandru.ivanus@lonlnx48:~/cm17q2> mm co -b cmapp/ngcmtrgr                           4:46PM
++ Initialising cmapp using hard linking
Checkout took 6.1 seconds
++ Updating dirstate for cmapp
++ Checking out ngcmtrgr
alexandru.ivanus@lonlnx48:~/cm17q2> gmake                                         4:47PM
++ Linking amd_root to /data/mwdeldev/ngdpack-14
++ Linking ngd_build_tools to /data/ngddeldev/int/ngd_build_tools/4-133-3-1
++ Linking build_monkey to /data/ngddeldev/int/jobmgr/2-7-0-9
++ Linking build_support to /data/ngddeldev/int/build_support/2-60-0
++ Linking ttserver to /data/mwdeldev/TTSERVER/2.6.2
++ Linking oracle to /remote/tools/Linux/2.6/oracle/products/11.2.0.4.4
++ Linking ngd_toolkit to /data/ngddeldev/int/ngd_toolkit/4-54-4-31
++ Linking ngd_test to /data/ngddeldev/int/ngd_test/5-18-2-15
++ Linking ngd_ttserver to /data/ngddeldev/int/ttserver/1-10-5ngd2
^Calexandru.ivanus@lonlnx48:~/cm17q2>                                               4:47PM
M
alexandru.ivanus@lonlnx48:~/cm17q2>
alexandru.ivanus@lonlnx48:~/cm17q2> which gmake                                4:47PM
gmake () {
    START_MAKE=`date +%s%3N`
    nice make -s -l15 -j20 $@
    RETURN_CODE=$?
    END_MAKE=`date +%s%3N`
    DIFF=$(echo "$END_MAKE-$START_MAKE" | bc)
    mkdir -p ~/.ngd_logs/grafana
    touch ~/.ngd_logs/grafana/stats.log
    CURRENT_PROJECT='test $NG_PROJECT = `ngdcs` && echo "CM" || echo "FM"'
    if [ $# -eq 0 ]
    then
        echo "make,make,gmake,`pwd -P`,$HOST,`date +"%Y-%m-%d %H:%M:%S.%3N"`,$CURRENT_PROJECT,$RETURN_CODE,null,$DIFF" >> ~/.ngd_logs/grafana/stats.log
    else
        echo "make,make,gmake $@,`pwd -P`,$HOST,`date +"%Y-%m-%d %H:%M:%S.%3N"`,$CURRENT_PROJECT,$RETURN_CODE,null,$DIFF" >> ~/.ngd_logs/grafana/stats.log
    fi
    echo "Test1.make.make.gmake.$HOST.$CURRENT_PROJECT.$RETURN_CODE.none:$DIFF|ms" | netcat -w 1 -u owl-statsd.etv.nce.amadeus.net 8125
    return $RETURN_CODE
}
alexandru.ivanus@lonlnx48:~/cm17q2> gmake                                         4:47PM
++ reading file cmapp/ngacc/Package.mk

```

Ilustración 8. Lanzando regresiones, comandos init y comprobacion de gmake

```
lonlnx48.lon.amadeus.net - PuTTY
++ Linking ngd_test to /data/ngddeldev/int/ngd_test/5-18-2-15
++ Linking ngd_ttserver to /data/ngddeldev/int/ttserver/1-10-5ngd2
^alexandru.ivanus@lonlnx48:~/cm17q2> 4:47PM
M
alexandru.ivanus@lonlnx48:~/cm17q2> alexandru.ivanus@lonlnx48:~/cm17q2> which gmake
gmake () {
    START_MAKE=`date +%s%3N`
    nice make -s -l15 -j20 $@
    RETURN_CODE=$?
    END_MAKE=`date +%s%3N`
    DIFF=$(echo "$END_MAKE-$START_MAKE" | bc)
    mkdir -p ~/.ngd_logs/grafana
    touch ~/.ngd_logs/grafana/stats.log
    CURRENT_PROJECT=test $NG_PROJECT = 'ngdcs' && echo "CM" || echo "FM"
    if [ $# -eq 0 ]
    then
        echo "make,make,gmake,`pwd -P`,$HOST,`date +"%Y-%m-%d %H:%M:%S.%3N"`,$CURRENT_PROJECT,$RETURN_CODE,null,$DIFF" >> ~/.ngd_logs/grafana/stats.log
    else
        echo "make,make,gmake $@,`pwd -P`,$HOST,`date +"%Y-%m-%d %H:%M:%S.%3N"`,$CURRENT_PROJECT,$RETURN_CODE,null,$DIFF" >> ~/.ngd_logs/grafana/stats.log
    fi
    echo "Test1.make.make.gmake.$HOST.$CURRENT_PROJECT.$RETURN_CODE.none:$DIFF|ms" | netcat -w 1 -u owl-statsd.etv.nce.amadeus.net 8125
    return $RETURN_CODE
}
alexandru.ivanus@lonlnx48:~/cm17q2> gmake 4:47PM
++ reading file cmapp/ngacc/Package.mk
++ reading file cmapp/ngaci/Package.mk
++ reading file cmapp/ngact/Package.mk
++ reading file cmapp/ngafw/Package.mk
++ reading file cmapp/ngams/Package.mk
++ reading file cmapp/ngasp/Package.mk
++ reading file cmapp/ngbzs/Package.mk
++ reading file cmapp/ngcds/Package.mk
++ reading file cmapp/ngcdt/Package.mk
++ reading file cmapp/ngcfa/Package.mk
++ reading file cmapp/ngcmb/Package.mk
++ reading file cmapp/ngcmt/Package.mk
++ reading file cmapp/ngcpm/Package.mk
++ reading file cmapp/ngcpr/Package.mk
++ reading file cmapp/ngcps/Package.mk
++ reading file cmapp/ngctp/Package.mk
++ reading file cmapp/ngetk/Package.mk
++ reading file cmapp/ngezt/Package.mk
++ reading file cmapp/ngpf/Package.mk
++ reading file cmapp/nghfw/Package.mk
++ reading file cmapp/nghst/Package.mk
++ reading file cmapp/ngiai/Package.mk
++ reading file cmapp/ngidc/Package.mk
++ reading file cmapp/nginv/Package.mk
++ reading file cmapp/nglcc/Package.mk
++ reading file cmapp/ngloy/Package.mk
++ reading file cmapp/nglst/Package.mk
++ reading file cmapp/ngmon/Package.mk
++ reading file cmapp/ngmpc/Package.mk
++ reading file cmapp/ngntr/Package.mk
++ reading file cmapp/ngonl/Package.mk
++ reading file cmapp/ngpag/Package.mk
++ reading file cmapp/ngpay/Package.mk
++ reading file cmapp/ngpcv/Package.mk
++ reading file cmapp/ngacc/Deps
```

Ilustración 9. Lanzando regresiones comando gmake

```

lonlnx48.lon.amadeus.net - PuTTY
=====
validating: cmrnd/ngbmv/src/bom/generate/yamls/BmvBulkMessageStats.yml
validating: cmrnd/ngbmv/src/bom/generate/yamls/BmvCustomizedElements.yml
validating: cmrnd/ngbmv/src/bom/generate/yamls/BmvElements.yml
validating: cmrnd/ngbmv/src/bom/generate/yamls/BmvExampleMessages.yml
validating: cmrnd/ngbmv/src/bom/generate/yamls/BmvMessage.yml
validating: cmrnd/ngbmv/src/bom/generate/yamls/BmvMessageStats.yml
validating: cmrlng/src/bom/generate/yamls/AcceptanceRecord.yml
validating: cmrlng/src/bom/generate/yamls/AdminHistory.yml
validating: cmrlng/src/bom/generate/yamls/CabinLevel.yml
validating: cmrlng/src/bom/generate/yamls/ConsolidatedStatistics.yml
validating: cmrlng/src/bom/generate/yamls/Flight.yml
validating: cmrlng/src/bom/generate/yamls/FqtvLevel.yml
validating: cmrlng/src/bom/generate/yamls/Lounge.yml
alexandru.ivanus@lonlnx48:~/cm17q2> ls                                         4:49PM
Makefile cmapp cmbmt cmcfg cmdba cmmwh cmsrv
cmapi cmapq cmbrd cmcore cmfnd cmrnd cmtec
alexandru.ivanus@lonlnx48:~/cm17q2> cd cmapp/ngcmtrgr                         4:57PM
alexandru.ivanus@lonlnx48:~/cm17q2/cmapp/ngcmtrgr> ls                           4:57PM
Deps Messages.csv TestPkgs edi try
alexandru.ivanus@lonlnx48:~/cm17q2/cmapp/ngcmtrgr> cd edi                      4:57PM
alexandru.ivanus@lonlnx48:~/cm17q2/cmapp/ngcmtrgr/edi> ntt conf                  4:57PM
++ using ngd_toolkit in .build (4-54-4-31)
+ exec /data/softs/bin/perl /data/vtmp/shadow/lonlnx48/remote/users/alexandru.ivanus/cm17q2/.build/n
gd_toolkit/bin/ngd_toolkit.pl conf
++ note: process running: 61887    29423 sei_master -tc mag.traceconf
++ note: process running: 61887    29512 sei_agent -nm 01_CML -tc agent.traceconf
++ note: process running: 61887    29724 otf_fe v14.0.0.43 : 01_CML FE_CML
++ note: process running: 61887    29770 otf_fe v14.0.0.43 : 01_CML LQS_CML
++ note: process running: 61887    29727 otf_cs v14.0.0.43 : 01_CML CS_FE_CML
++ note: process running: 61887    29731 otf_be v14.0.0.43 : 01_CML LCK_LCK_CML #1 [idle]
++ note: process running: 61887    29735 otf_be v14.0.0.36 : 01_CML ETS_RIC_CML #1 [idle]
++ note: process running: 61887    29738 otf_be v14.0.0.43 : 01_CML GMC_ABRCache_CML #1 [idle]
++ note: process running: 61887    29741 otf_be v14.0.0.43 : 01_CML GMC_EZT-RFDSynchro_CML #1 [idle]
++ note: process running: 61887    29758 otf_be v14.0.0.43 : 01_CML GMC_EZT-FMTSyncro_CML #1 [idle]
++ note: process running: 61887    29760 otf_be v14.0.0.43 : 01_CML GMC_EZT-EZTSynchro_CML #1 [idle]
++ note: process running: 61887    29763 otf_be v14.0.0.43 : 01_CML CSC_Cache_CML #1 [idle]
++ note: process running: 61887    29767 otf_be v14.0.0.43 : 01_CML COM_AQM_CML #1 [idle]
ntt: should not try to build the configuration while servers are running
alexandru.ivanus@lonlnx48:~/cm17q2/cmapp/ngcmtrgr/edi> ntt vsy                                         4:57PM
++ using ngd_toolkit in .build (4-54-4-31)
+ exec /data/softs/bin/perl /data/vtmp/shadow/lonlnx48/remote/users/alexandru.ivanus/cm17q2/.build/n
gd_toolkit/bin/ngd_toolkit.pl vsy
++ note: process running: 61887    29423 sei_master -tc mag.traceconf
++ note: process running: 61887    29512 sei_agent -nm 01_CML -tc agent.traceconf
++ note: process running: 61887    29724 otf_fe v14.0.0.43 : 01_CML FE_CML
++ note: process running: 61887    29770 otf_fe v14.0.0.43 : 01_CML LQS_CML
++ note: process running: 61887    29727 otf_cs v14.0.0.43 : 01_CML CS_FE_CML
++ note: process running: 61887    29731 otf_be v14.0.0.43 : 01_CML LCK_LCK_CML #1 [idle]
++ note: process running: 61887    29735 otf_be v14.0.0.36 : 01_CML ETS_RIC_CML #1 [idle]
++ note: process running: 61887    29738 otf_be v14.0.0.43 : 01_CML GMC_ABRCache_CML #1 [idle]
++ note: process running: 61887    29741 otf_be v14.0.0.43 : 01_CML GMC_EZT-RFDSynchro_CML #1 [idle]
++ note: process running: 61887    29758 otf_be v14.0.0.43 : 01_CML GMC_EZT-FMTSyncro_CML #1 [idle]
++ note: process running: 61887    29760 otf_be v14.0.0.43 : 01_CML GMC_EZT-EZTSynchro_CML #1 [idle]
++ note: process running: 61887    29763 otf_be v14.0.0.43 : 01_CML CSC_Cache_CML #1 [idle]
++ note: process running: 61887    29767 otf_be v14.0.0.43 : 01_CML COM_AQM_CML #1 [idle]
++ !! Warning !! : Could not run config for vsy but attempting to kill left over sei processes
++ !! Warning !! : Remaining sei_agent process (pid 29512), killing it
++ killing spawned sei_agent process on PID 29512 + child processes
++ !! Warning !! : Remaining sei_master process (pid 29423), killing it
++ killing spawned sei_master process on PID 29423 + child processes
++ !! Warning !! : Remaining otf_fe process (pid 29724), killing it
++ killing spawned otf_fe process on PID 29724 + child processes
++ !! Warning !! : Remaining otf_fe process (pid 29770), killing it

```

Ilustración 10. Lanzando regresiones comandos ntt conf y ntt vsy

```

lonlnx48.lon.amadeus.net - PuTTY
2017/11/23 17:02:05.539101 cliBuilder-20023 <TraceMgrImpl.cpp#538> TRACER DEBUG - Trace output thread joint
2017/11/23 17:02:05.543101 cliBuilder-20023 <TraceMgrImpl.cpp#538> TRACER DEBUG - Trace output thread joint
In case of failure, logs will be available in the latest /gctmp/alexandru.ivanus/cliBuilder file
Old type versioners are deprecated and should be replaced by strong ones (versioner mdw::Pack from mdw::ABRCClient::tools::cacheBuilder 13-0-3-2).
Old type versioners are deprecated and should be replaced by strong ones (versioner osp::OSPack from mdw::ABRCClient::tools::cacheBuilder 13-0-3-2).
mdw::Pack 13-0-2-11 is deprecated and should not be used : 08618703 HPTimer 13-0-0-0 and 13-0-0-1 can lead to core with OTF
BLACKLISTED.
mdw::DeviceCache::api 13-0-0-1 is deprecated and should not be used : PTR 07938088 [Medium]: WGR mode: The PNR is redisplayed even if it is in WGR.
Launching /gctmp/alexandru.ivanus/tmpYIVQql/bin/Linux2-6_64/MultiThread/Debug/g++_4_3_2/cliBuilder -s /data/vtmp/ngdcbsld/PRD_2018Q2/58-0-15/internal/cmcore/ngcomrgr/tools/rules/ruletypes.xml -o 1A -a DCS -m 1 -r -u
2017/11/23 17:02:09.895222 cliBuilder-27235 <ConfigManager.cpp#117> TRACER INFO - Reading configuration from configuration file
2017/11/23 17:02:09.899223 cliBuilder-27235 <ConfigManager.cpp#147> TRACER INFO - Loading dynamic configuration
2017/11/23 17:02:09.915223 cliBuilder-27235 <TraceMgrImpl.cpp#443> TRACER DEBUG - Trace output thread spawned
2017/11/23 17:02:41.048094 cliBuilder-27235 <TraceMgrImpl.cpp#497> TRACER DEBUG - Output queue flushed
2017/11/23 17:02:41.048094 cliBuilder-27235 <TraceMgrImpl.cpp#498> TRACER DEBUG - Trace output thread exiting
2017/11/23 17:02:41.048094 cliBuilder-27235 <TraceMgrImpl.cpp#538> TRACER DEBUG - Trace output thread joint
2017/11/23 17:02:41.048094 cliBuilder-27235 <TraceMgrImpl.cpp#538> TRACER DEBUG - Trace output thread joint
alexandru.ivanus@lonlnx48:~/cm17q2/cmapp/ngcmtrgr/edi> ntt loadbasedb
++ using ngd_toolkit in .build (4-54-4-31)
+ exec /data/softs/bin/perl /data/vtmp/shadow/lonlnx48/remote/users/alexandru.ivanus/cm17q2/.build/ngd_toolkit/bin/ngd_toolkit.pl loadbasedb
++ Multi Peak user:alexandruivanus server:NCMUSR04LS
++ ORACLE_HOME=/nastools/oracle/products/11.2.0.4.4 /bin/sh /data/vtmp/shadow/lonlnx48/remote/users/alexandru.ivanus/cm17q2/cmcore/ngcomrgr/tools/db/scripts/loadbasedb.sh -u alexandruivanus -p alexandruivanus -s NCMUSR04LS
++ Using QDM
++ check '/data/vtmp/shadow/lonlnx48/remote/users/alexandru.ivanus/cm17q2/cmapp/ngcmtrgr/edi/log/loadbasedb-alexandruivanus.log' for specific messages
UserInfo: alexandruivanus alexandruivanus NCMUSR04LS
++ Dropping 'alexandruivanus' schema
++ Importing regression database
--> using dump files '/data/vtmp/shadow/lonlnx48/remote/users/alexandru.ivanus/cm17q2/cmdb/ngdba/dumps/default/current_cm'
++ creating dbrefresh helper-objects (flashback)
++ Multi Peak user:alexandruivanus_p2 server:NCMUSR04LS
++ ORACLE_HOME=/nastools/oracle/products/11.2.0.4.4 /bin/sh /data/vtmp/shadow/lonlnx48/remote/users/alexandru.ivanus/cm17q2/cmcore/ngcomrgr/tools/db/scripts/loadbasedb.sh -u alexandruivanus_p2 -p alexandruivanus_p2 -s NCMUSR04LS
++ Using QDM
++ check '/data/vtmp/shadow/lonlnx48/remote/users/alexandru.ivanus/cm17q2/cmapp/ngcmtrgr/edi/log/loadbasedb-alexandruivanus_p2.log' for specific messages
UserInfo: alexandruivanus_p2 alexandruivanus_p2 NCMUSR04LS
++ Dropping 'alexandruivanus_p2' schema
++ Importing regression database
--> using dump files '/data/vtmp/shadow/lonlnx48/remote/users/alexandru.ivanus/cm17q2/cmdb/ngdba/dumps/default/current_cm'
++ creating dbrefresh helper-objects (flashback)
++ /bin/sh /data/vtmp/shadow/lonlnx48/remote/users/alexandru.ivanus/cm17q2/cmcore/ngcomrgr/tools/db/scripts/multischema_grants.sh alexandruivanus alexandruivanus NCMUSR04LS alexandruivanus_p2

```

Ilustración 11. Lanzando regresiones, comando loadbasedb

```

lonlnx48.lon.amadeus.net - PuTTY
With the Partitioning, Oracle Label Security, OLAP, Data Mining,
Oracle Database Vault and Real Application Testing options

SQL> SQL> 2 3 4 5 6 7 8 9 10 11 12 13
PL/SQL procedure successfully completed.

SQL> Disconnected from Oracle Database 11g Enterprise Edition Release 11.2.0.2.0 - 64bit Production
With the Partitioning, Oracle Label Security, OLAP, Data Mining,
Oracle Database Vault and Real Application Testing options
Granted all alexandruivanus_p2 -> alexandruivanus
alexandruivanus@lonlnx48:~/cm17q2/cmapp/ngcmtrgr/edi> ntt boot 5:05PM
++ using ngd_toolkit in .build (4-54-4-31)
+ exec /data/softs/bin/perl /data/vtmp/shadow/lonlnx48/remote/users/alexandruivanus/cm17q2/.build/ngd_toolkit/bin/ngd_toolkit.pl boot
++ Notice: OTF Front End is at: 172.29.7.34, port 20103
++ Starting the master agent
++ Waiting for master agent to be ready (on 172.29.7.34:20113)
++ sei_admin -c root password stop as 01_CML -F
Connect to a master agent via the following address and port: 172.29.7.34:20113
Establishing secured connection for user root
We are connected to a master agent.

<<< LOGIN USER root (executed 2017/11/23 at 17:07:30.004)

Ok.

<<< STOP APPLICATION SERVER 01_CML (executed 2017/11/23 at 17:07:30.004)

Ok.

++ sei_admin -c root password start as 01_CML -F
Connect to a master agent via the following address and port: 172.29.7.34:20113
Establishing secured connection for user root
We are connected to a master agent.

<<< LOGIN USER root (executed 2017/11/23 at 17:07:30.080)

Ok.

<<< START APPLICATION SERVER 01_CML (executed 2017/11/23 at 17:07:30.080)

Ok.

++ Starting the local agent
++ front-end stopped...
++ front-end configuring...
++ front-end configuring...
++ front-end ready
++ Starting the receptor router (will simply forward the queries to the receptor)
++ Checking if TTS receptor is already running on 172.29.7.34:201XX
++ Running sudo /usr/bin/lsof -nPi | grep 172.29.7.34:201 | grep vsantiago | grep -w TTServer | awk '{print $2;}' 
++ Executing: PYTHONPATH=/data/softs/python/install/lib/python2.4/site-packages /data/mwdeldev/TTSERVER/last-version/scripts/TTServer.sh -s etc/router.py > log/Router.log 2>&1 on PID 3961
++ starting back-ends
++ RGR_TCH configuring...
++ RGR_TCH ready
++ RGR_CPO ready
++ RGR_INT ready
++ RGR_INT ready
++ RGR_SMB ready

```

Ilustración 12. Se ejecuta ntt boot

```

alexandruivanus@lonlnx48:~/cm17q2/cmapp/ngcmtrgr/edi> ntt t 5:07PM
++ using ngd_toolkit in .build (4-54-4-31)
+ exec /data/softs/bin/perl /data/vtmp/shadow/lonlnx48/remote/users/alexandruivanus/cm17q2/.build/ngd_toolkit/bin/ngd_toolkit.pl t
++ using ngd_test in .build (5-18-2-15)
++ using ttserver from .build (1-10-5ngd2)
--- Processing cache ---
--- Processing CITMAQ ---
Refreshing database
Time spent refreshing the database: 2.0 sec
CheckHandlingActivations.Case_001.scn OK (0.5 sec)
CheckHandlingActivations.Case_002.scn OK (0.1 sec)
CheckHandlingActivations.Case_003.scn OK (0.1 sec)

```

Ilustración 13. Se ejecuta ntt t lanzando escenarios y obteniendo resultados.

```

alexandru.ivanus@lonlnx48:~/cm17q2/cmapp/ngcmtrgr/edi> ntt PROFILER=oprofile config      5:24PM
++ using ngd_toolkit in .build (4-54-4-31)
+ exec /data/softs/bin/perl /data/vtmp/shadow/lonlnx48/remote/users/alexandru.ivanus/cm17q2/.build/ngd_toolkit/bin/ngd_toolkit.pl PROFILER=oprofile config
++ loading ntt.conf
++ loading /data/vtmp/shadow/lonlnx48/remote/users/alexandru.ivanus/cm17q2/cmcore/ngcomrgr/tools/lib/defaults.ngt
++ getting configuration settings from makefile
++ getting configuration settings from makefile
readlink: missing operand
Try `readlink --help' for more information.
++ user.conf: /projects/ngdcs/users.conf
++ Port offset is 51
++ Command to run /data/ngddeldev/int/util/default/bin/lock_port.sh l -fjson -m 20100 -M 20199
++ Command to run /data/ngddeldev/int/util/default/bin/lock_port.sh r -p 20100 --force-lock --lock-name=.port.lock.alexandru.ivanus --lock-folder=/data/vtmp/shadow/lonlnx48/remote/users/alexandru.ivanus/cm17q2 -fjson -M 45500
++ Lock process successfull
++ dbinfo_pk2: alexandruivanus_p2/alexandruivanus_p2/NCMUSR04LS
++ Calling make to create /data/vtmp/shadow/lonlnx48/remote/users/alexandru.ivanus/cm17q2/.delivery.alexandru.ivanus
++ make -s AMD_DEBUG=1 VERSION_NAME2=NGD_0.0.0 install INSTALL_DIR="/data/vtmp/shadow/lonlnx48/remote/users/alexandru.ivanus/cm17q2/.delivery.alexandru.ivanus"
Warning in cmapp/nglcc/Package.mk :
Servers are defined but no install component found!
Warning in cmrnd/ngbds/Package.mk :
Servers are defined but no install component found!
Warning in cmtc/ngweb/Package.mk :
Servers are defined but no install component found!
Adding external component CML-MHI-14.0.3.0.tar.gz
Adding external component CML-MHA-14.0.0.0.tar.gz
Adding external component CML-ETS-14.1.0.0.tar.gz
++ Installing the MDW & NGD libraries
++ Validating canned messages File: /projects/ngdcsbld/pwe/ngtools/rfd/data/crb_canned
++ Validating canned messages File: /projects/ngdcsbld/pwe/ngtools/rfd/data/crb_canned_v2
++ Installing binaries for com
++ Installing libraries for com
++ Installing libraries for com
++ Installing libraries for mhi
++ Installing conf for mhi
++ Installing deployment.xml for mhi
++ Installing libraries for mha
++ Installing conf for mha
++ Installing deployment.xml for mha
++ Installing libraries for ets
++ Installing deployment.xml for ets
++ Installing libraries for csc
++ Installing libraries for csp
++ Installing libraries for acc
++ Installing libraries for aci
++ Installing libraries for act
++ Installing libraries for afw
++ Installing libraries for afwtst
++ Installing libraries for ams
++ Installing libraries for amstst
++ Installing libraries for asp
++ Installing libraries for bsr
++ Installing libraries for cds
++ Installing libraries for cdt

```

Ilustración 14. Ejecutando Profiler

```

anu/www/deployment.xml
++ Copy the com data
++ Installing databases configuration
++ Fetching static resources
++ Creating setenv file in /data/vtmp/shadow/lonlnx48/remote/users/alexandru.ivanus/cm17q2/.delivery.alexandru.ivanus/setenv
Start of Merged BE Preparation
++ Launchng /data/vtmp/shadow/lonlnx48/remote/users/alexandru.ivanus/cm17q2/.build/ngd_toolkit/bin/..../lib/NGD_Toolkit/DeployMerge/.../pylibs/bin/merge_deploy.sh -i /data/vtmp/shadow/lonlnx48/remote/users/alexandru.ivanus/cm17q2/.delivery.alexandru.ivanus -b TCH,CPO,INT,INT,SMB,HST,FLI,CPR,IDC,SYN,ACR -m /data/vtmp/shadow/lonlnx48/remote/users/alexandru.ivanus/cm17q2/cmapp/ngcmtrgr/edi -o /data/vtmp/shadow/lonlnx48/remote/users/alexandru.ivanus/cm17q2/cmapp/ngcmtrgr/edi/cml/rgr/deployment.xml for merge
++ Merge finished. Output:
End of Merged BE Preparation
++ Preparing and filtering /data/vtmp/shadow/lonlnx48/remote/users/alexandru.ivanus/cm17q2/.delivery.alexandru.ivanus/cfg/etc/configuration.xml to cml/cfg/etc/configuration.xml
++ writing to etc/implementation.xml
++ writing to etc/technical.xml
++ writing to etc/mag.traceconf
++ writing to etc/agent.traceconf
++ writing to etc/FF.traceconf
++ writing to etc/LQS.traceconf
++ writing to etc/CS_FE.traceconf
++ writing to etc/RGR_TCH.traceconf
++ writing to etc/RGR_CPO.traceconf
++ writing to etc/RGR_INT.traceconf
++ writing to etc/RGR_INT.traceconf
++ writing to etc/RGR_SMB.traceconf
++ writing to etc/RGR_HST.traceconf
++ writing to etc/RGR_FLI.traceconf
++ writing to etc/RGR_CPR.traceconf
++ writing to etc/RGR_IDC.traceconf
++ writing to etc/RGR_SYN.traceconf
++ writing to etc/RGR_ACR.traceconf
++ copying reference PCV cache to data directory
++ Getting PCV regressions data from ngcomrgr
++ writing to etc/setenv
++ writing to etc/router.py
++ writing to etc/ola_patch.pl
++ writing to etc/otf_config.sh
++ creating the OTF configuration
chmod: changing permissions of `/data/ngddeldev/int/ngd_toolkit/4-54-4-31/pylibs/bin/..../ntt_template/ntt_template.py': Operation not permitted
++ Template generated successfully from /data/vtmp/shadow/lonlnx48/remote/users/alexandru.ivanus/cm17q2/.build/ngd_toolkit/bin/..../lib/NGD_Toolkit/.../pylibs/ntt_template/templates/ngd_testrc.tpl
++ ngdtest_rc generated successfully
++ peak1:
++ peak2: BA
++ writing to /data/vtmp/shadow/lonlnx48/remote/users/alexandru.ivanus/cm17q2/cmapp/ngcmtrgr/edi/etc/peakdata.xml
++ DB_user_p1: alexandruivanus
++ DB_user_p2: alexandruivanus_p2
++ writing to /data/vtmp/shadow/lonlnx48/remote/users/alexandru.ivanus/cm17q2/cmapp/ngcmtrgr/edi/etc/dbconnectors.xml
++ etc/otf_config.sh
Starting the master agent
## OLA configure ##
## OLA implement ##
## OLA activate:lqs ##
## OLA activate:otf ##
## OLA activate:rgr ##
alexandru.ivanus@lonlnx48:~/cm17q2/cmapp/ngcmtrgr/edi> 5:27PM

```

Ilustración 15. Proceso finalizado con éxito

Tras tener preparados los sets, se proceder a lanzar las regresiones localmente con Operf[6] como se describe a continuación en el siguiente apartado.

Para comprobar que correctamente se ha lanzado la regresión únicamente hay que consultar la siguiente página registrándose siempre con el unix id que corresponda como se muestra en la captura de pantalla.

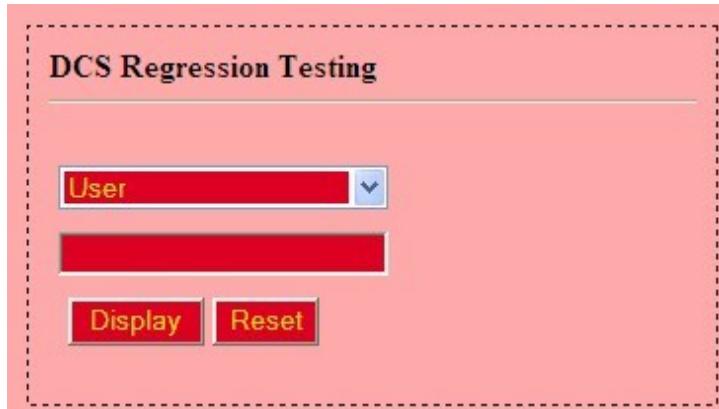


Ilustración 16. Ingreso a la dashboard donde se visualizan las regresiones lanzadas por los usuarios.

Una vez se haya ingresado o simplemente consultado las regresiones ejecutadas por un usuario (en este caso consultamos las realizadas por el usuario *rchossar* el día 24 de marzo del año 2010 por ejemplo), podremos ver si fallaron o si los resultados fueron satisfactorios.

En este caso se puede ver que fallaron para la lista de escenarios que se muestran al igual que también se pueden ver los logs y el tiempo que consumieron.

NGDCS User Regression Report for user rchossar 24-03-2010						
>>run >>branch >>release >>package >>type >>function						
			Duration	OK	Total	%
> RUN	Regression		17m 9s	0	162	0 %
>> BRANCH	DEV		17m 9s	0	162	0 %
>>> RELEASE	27-0-202		17m 9s	0	162	0 %
>>>> PACKAGE	smt		17m 9s	0	162	0 %
>>>>> TYPE	cry		6m 36s	0	19	0 %
>>>>> FUNCTION	STO		6m 36s	0	19	0 %
Scenario		Status	Logs	Duration	Time	User
RetrieveSeatsOccupied.Case_001.scn	[input]	Failed	[log][diff]	20.2 s	14:53:50	rchossar
RetrieveSeatsOccupied.Case_002.scn	[input]	Failed	[log][diff]	20.1 s	14:54:11	rchossar
RetrieveSeatsOccupied.Case_003.scn	[input]	Failed	[log][diff]	20.2 s	14:54:31	rchossar
RetrieveSeatsOccupied.Case_004.scn	[input]	Failed	[log][diff]	20.3 s	14:54:51	rchossar
RetrieveSeatsOccupied.Case_005.scn	[input]	Failed	[log][diff]	20.2 s	14:54:52	rchossar
RetrieveSeatsOccupied.Case_006.scn	[input]	Failed	[log][diff]	24.7 s	14:55:16	rchossar
RetrieveSeatsOccupied.Case_007.scn	[input]	Failed	[log][diff]	24.4 s	14:55:41	rchossar
RetrieveSeatsOccupied.Case_008.scn	[input]	Failed	[log][diff]	23.5 s	14:56:24	rchossar
RetrieveSeatsOccupied.Case_009.scn	[input]	Failed	[log][diff]	20.3 s	14:56:45	rchossar
RetrieveSeatsOccupied.Case_010.scn	[input]	Failed	[log][diff]	20.4 s	14:56:45	rchossar
DatosCantidadesEnAnt.scn	[input]	Failed	[log][diff]	20.1 s	14:57:25	rchossar

Ilustración 17. Ejemplos de ejecuciones ejecutadas por un usuario.

NGDCS CM Regression Report for DEV (27-0-211)				
>>branch >>release >>package >>type >>function				
		Duration	OK	Total
> BRANCH	DEV	11h 34m 52s	21011	22816
>> RELEASE	27-0-211	11h 34m 52s	21011	22816
>>> PACKAGE	acc	23m 30s	1406	1417
>>> PACKAGE	bmt	44m 40s	2573	2861
>>> PACKAGE	brd	12m 2s	1256	1268
>>> PACKAGE	cds	3s	26	26
>>> PACKAGE	cfa	7m 4s	207	217
>>> PACKAGE	cmt	9m 32s	127	132
>>> PACKAGE	cpr	24m 7s	2032	2061
>>> PACKAGE	cry	2m 23s	36	41
>>> PACKAGE	ctp	50m 27s	874	988
>>> PACKAGE	ehb	4s	21	21
>>> PACKAGE	etk	19m 56s	618	629
>>> PACKAGE	fli	15m 4s	846	950
>>> PACKAGE	fmc	0s	0	238
>>> PACKAGE	hst	2m 58s	198	224
>>> PACKAGE	hub	10s	7	8
>>> PACKAGE	idc	3m 25s	683	695
>>> PACKAGE	inv	4m 20s	53	106
>>> PACKAGE	lst	21m 19s	1871	1953
>>> PACKAGE	onl	23m 31s	1310	1345
>>> PACKAGE	pag	26s	57	58
>>> PACKAGE	prt	24m 45s	1219	1223
>>> PACKAGE	reg	24m 25s	906	1001
>>> PACKAGE	rgd	3m 52s	290	337
>>> PACKAGE	sec	1m 15s	109	129
>>> PACKAGE	smt	1h 30m 18s	1707	1986
>>> PACKAGE	swp	0s	82	82
>>> PACKAGE	syn	2h 40m 5s	1117	1411
>>> PACKAGE	tci	2h 4m 57s	1380	1409

Ilustración 18. Ejemplo de las regresiones lanzadas en la rama DEV.

También se pueden por ejemplo consultar las regresiones lanzadas y el tiempo que tardaron y el porcentaje de completitud de cada set de regresiones.

Un ejemplo se tiene en la imagen anterior la cual nos esta mostrando dentro de la rama llamada DEV, en su release 27-0-211 para todos los paquetes de regresiones.

Como se puede observar el proceso tardo mas de 11 horas y 34 minutos ya que en este tiempo llevaba el 92% de completitud.

Por lo que como se puede ver se pierde mucho tiempo ya que se podría ejecutar simplemente el paquete de regresiones que afecte directamente al código el cual probablemente tardaría desde segundos hasta dos o tres horas en vez de ejecutar toda la suite la cual tarda mucho más.

2.1.2. Entorno de las dashboards

Las dashboards del departamento Software Framework Architecture (SFA) en el cual se desarrolla este proyecto están localizadas en un repositorio de Bitbucket llamado INT_DASHBOARDS en el cual se va a crear la nueva dashboard la cual nos permita crear nuevos usuarios de regresión los cuales puedan ejecutar los test de regresión necesarios para testear el proyecto como se ha mencionado anteriormente en el punto 2.1.1.

Es necesario crear estos usuarios dándoles un puerto en el que trabajar, y creándolos en una de las bases de datos del entorno específico en el que se van a ejecutar los test de regresión.

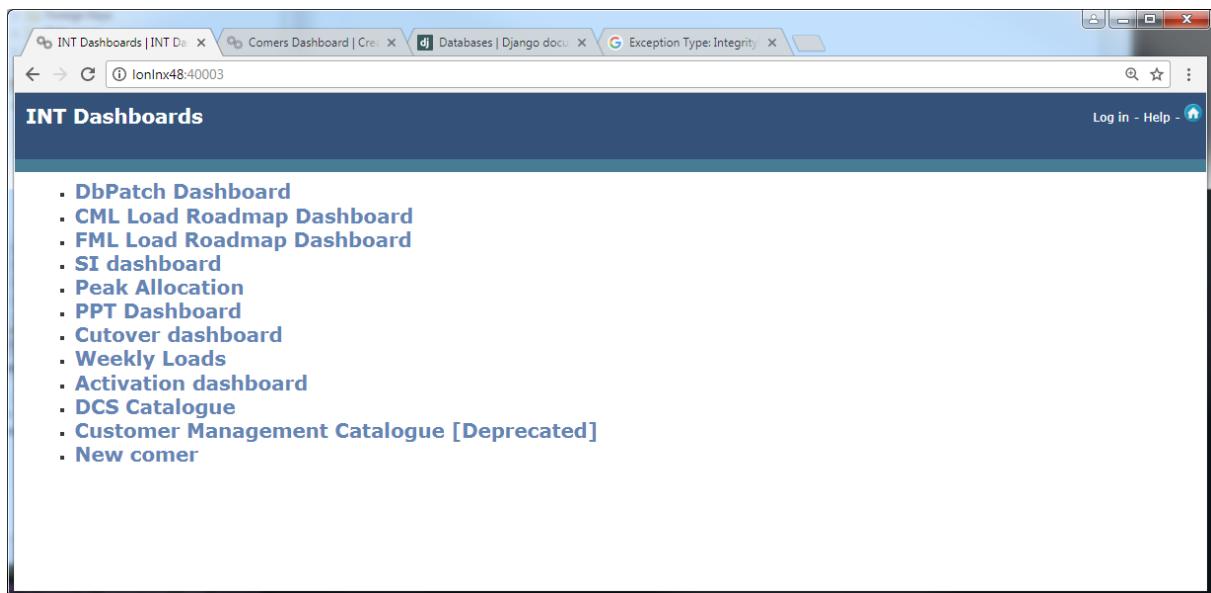


Ilustración 19. Imagen del panel principal de las dashboards

3. Análisis de requisitos

3.1. Propósito

El principal propósito de este proyecto es básicamente a partir de un conjunto de ficheros del código del producto modificados, establecer que test de regresión han de ser ejecutados minimizando con ello el coste.

3.2. Ámbito del Sistema

El sistema en si resulta un tanto complejo puesto que esta formado por múltiples entornos, proyectos y módulos que configuran el producto final, por lo que inicialmente este proyecto va a estar enfocado únicamente en la sección Customers Management Leaves o CML específicamente en uno de sus módulos.

Este sistema en general va a estar comprendido por tres tipos de usuario: usuario administrador o gestor el cual dará de alta por medio de la aplicación web de Django[7] a los usuarios que podrán ejecutar las regresiones y usuario de carga y ejecución de regresiones.

3.3. Descripción general

3.3.1. Perspectiva del producto

Se pretende que el producto alcance un uso extendido tras su puesta en marcha puesto que se prevé que facilitara la labor de testeo del producto en un corto plazo por lo cual reducirá los costes de tiempo y permitirá agilizar el procedimiento requerido para la obtención del producto final.

3.3.2. Funciones del producto

A continuación, se van a detallar las funciones esperadas dividiendo este proyecto en pequeños módulos funcionales independientes.

Módulo 1

Este modulo se corresponderá con el modulo encargado de transformar los distintos escenarios que configuran los test de regresión los cuales abarcan cada subsistema.

Se llevará a cabo la generación de ficheros csv los cuales nos indicaran por cada subset de regresiones las funciones que testean, las dependencias entre ellas y las bibliotecas que usan.

Como requisito a cumplir, el principal cometido a realizar en este módulo será crear los distintos ficheros csv en los cuales se almacene la información correspondiente a cada escenario.

Módulo 2

En este modulo se trataran las partes correspondientes al código del producto final de la compañía o sus subproductos en los cuales se van a realizar estas modificaciones de código las cuales queremos testear con los test de regresión que se tienen especificado para los distintos módulos.

La principal funcionalidad que ha de cumplir este módulo ha de ser la de partiendo de un conjunto de ficheros de código c++ se han de extraer las funciones que se hallan en cada uno de los ficheros que han sido modificados, es decir, se han de conocer los cambios que se han realizado en el código para llevar a cabo la extracción de las funciones que se hayan implicadas en estos cambios las cuales van a ser almacenadas en una base de datos o fichero.

Módulo 3

Dicho modulo comprenderá la parte propia del mapping en el cual se ha de obtener finalmente cada subset de test de regresión a ejecutar dependiendo de los resultados que se obtengan de las modificaciones del código c++ del Módulo 2 y los subset de escenarios de regresión del Módulo 1.

Módulo 4

Se corresponde con la aplicación web creada con el fin de añadir nuevos usuarios al sistema para poder ejecutar regresiones en el sistema.

Se ha de mencionar que este modulo en la practica puede ser un poco mas complicado de lo que parece puesto que no resulta ser una simple aplicación web CRUD en la cual el administrador pueda crear, actualizar, visualizar y eliminar usuarios sino que también se han de manejar donde se han de crear estos usuarios ya que actualmente se cuentan con dos bases de datos distintas localizadas en distintas partes del mundo, la primera en Sydney y la segunda en Londres. Ademas se ha de mencionar que cada base de datos alberga los distintos entornos de los proyectos por lo que ademas se ha de especificar cuando se añada el usuario este entorno en el que va a poder lanzar regresiones.

También como requisito fundamental de este modulo, se ha de crear una asignación inteligente de puertos a cada usuario nuevo que se ingrese. Otra restricción a tratar consiste en que el usuario a ingresar tiene tener un Unix Id propio el cual haga único a cada usuario. Este Unix Id debe de ser el mismo que el que el usuario usará en los demás sistemas evitando así el tener un usuario identificado con varios id.

Otro requisito requerido consiste en que la aplicación ha de tener la opción de migrar los usuarios que se seleccionen de una base de datos a otra y de un entorno a otro sin la necesidad de eliminar el usuario y crearlo de nuevo. Unicamente se requiere un checkbox el cual pueda seleccionar los usuarios que se quieran migrar.

También se ha de implementar una zona de búsqueda general y búsqueda por campos la cual permita rápidamente obtener los resultados y datos correspondientes a un usuario, un puerto, una base de datos, un entorno, etc.

Se han de llevar a cabo todas las sincronizaciones y despliegue en las maquinas en las que se trabaja, al menos en el host lonlnx48 el cual esta dedicado a integración y control de las bases de datos entre otras funcionalidades.

Además, puesto que el ingreso de usuarios se viene haciendo manualmente aplicando estos cambios directamente sobre una base de datos e ingresando el usuario en un fichero llamado users.conf localizado en Bitbucket, se han de

gestionar el posible manejo de este fichero por usuarios que no conozcan la aplicación o simplemente como alternativa a la aplicación en caso de que alguna vez dejase de funcionar.

Por lo que cada vez que se vaya a usar la aplicación simplemente se ha de asegurar que no se han ingresado nuevos usuarios en dicho fichero o simplemente se ha de hacer una importación de este fichero a nuestra base de datos comprobando así que no hay ningún usuario nuevo, y si lo hay se ha de ingresar automáticamente.

Otro requisito consiste en que se ha de poder descargar esta información de usuarios existentes en las bases de datos desde su inicio en un fichero para visualizar por ejemplo en Excel.

3.3.3. Restricciones

Como restricciones se cuenta con que nuestra base de código está en C ++ y nuestras pruebas de regresión están escritas en Perl.

Para el almacenamiento de cualquier información, tenemos una variedad de opciones disponibles que van desde opciones convencionales de SQL como Oracle y MySQL o más novedosas opciones NoSQL como MongoDB, Elasticsearch o alguna solución de base de datos de gráficos.

En términos de lenguaje de desarrollo, la opción requerida es Python. Para recopilar la información bruta, también será necesario usar algunos perfiles los cuales vendrán dados por herramientas que ya viene usando la compañía como bien son Callgrind y operf como opciones viables en este proyecto.

Para la aplicación web se ha de usar Django puesto que actualmente las demás aplicaciones web y el sistema en el que se va a integrar están conformados en una aplicación web Django de dashboards, ademas otra restricción es que la base de datos es SQL por lo que se usarán herramientas acorde a ella.

3.3.4. Suposiciones y dependencias

Finalmente, una vez el proyecto este funcionando y sea usado, dependerá explícitamente del mantenimiento que le proporcionen el personal del equipo de Software Framework Architecture (SFA) en el cual se esta desarrollando este proyecto.

Este equipo sera el encargado de la carga/creación de los usuarios de regresión (usuarios con los permisos necesarios para ejecutar unas determinadas regresiones en un entorno) con el fin de poder agilizar el procedimiento de pruebas del nuevo código.

3.4. Requisitos Específicos

A continuación en este apartado se va a detallar con mayor profundidad los requisitos necesarios siguiendo el siguiente esquema:

- 1. Requisitos funcionales**
- 2. Requisitos de rendimiento**
- 3. Requisitos tecnológicos**
- 4. Atributos del sistema**
 - 1. Requisitos de seguridad**
 - 2. Requisitos de fiabilidad y disponibilidad**
 - 3. Requisitos de implementación**
- 5. Otros requisitos**
 - 1. Requisitos económicos**
 - 2. Requisitos de accesibilidad**
 - 3. Requisitos futuros**

1. Requisitos funcionales

NOMBRE	RF1 - USUARIOS DE LA APLICACIÓN WEB
FUENTE	ANALISTA
DESCRIPCIÓN	<p>La aplicación Web de registro y consulta de usuarios de regresión va a tener dos tipos de usuario:</p> <ul style="list-style-type: none"> • Usuario Gestor/Administrador • Usuario de Consulta
PRIORIDAD	ALTA/NECESARIO

NOMBRE	RF2 - USUARIO ADMINISTRADOR DE LA APLICACIÓN WEB
FUENTE	ANALISTA
DESCRIPCIÓN	<p>El gestor es el responsable de la administración y podrá llevar a cabo todas las acciones posibles.</p> <ul style="list-style-type: none"> • Insertar nuevos usuarios. • Modificar usuarios ya existentes. • Eliminar usuarios. • Migrar usuarios. • Asignar puertos. • Asignar nombre de usuario en una determinada base de datos. • Realizar sincronizaciones. • Listar usuarios. • Generar fichero csv con el listado de los usuarios.
PRIORIDAD	ALTA/NECESARIO

NOMBRE	RF3 - USUARIO DE CONSULTA DE LA APLICACIÓN WEB
FUENTE	ANALISTA
DESCRIPCIÓN	El usuario de carga realizará únicamente las operaciones referentes a la consulta de usuarios registrados (id, puerto asignado, base de datos, entorno, nombre, etc)
PRIORIDAD	ALTA/NECESARIO

NOMBRE	RF4 - INTERFAZ EN INGLÉS
FUENTE	ANALISTA
DESCRIPCIÓN	La aplicación y la herramienta de tests deben de estar soportadas en el idioma internacional.
PRIORIDAD	ALTA / NECESARIO

NOMBRE	RF5 - SIMPLIFICACIÓN DE LA INTERACCIÓN HUMANA
FUENTE	ANALISTA
DESCRIPCIÓN	La interfaz con la que interactúan los usuarios debe de ser lo más amigable y simple posible. Su fin debe de ser que el usuario pueda manejarla sin tener conocimientos previos, de forma sencilla, cómoda e intuitiva tanto en la aplicación de creación de usuarios como en la herramienta de testeo.
PRIORIDAD	ALTA / NECESARIO

NOMBRE	RF6 - INTERFACES ELEMENTALES
FUENTE	ANALISTA
DESCRIPCIÓN	La interacción que realice el usuario sobre la aplicación web y sobre la herramienta de tests debe de ser simple, de pocos pasos y sin abrumar al usuario. Su fin debe de ser que el usuario pueda manejarla sin tener conocimientos previos, de forma sencilla, cómoda e intuitiva sin realizar demasiadas iteraciones para llegar a realizar las acciones que se deseé.
PRIORIDAD	ALTA/NECESARIO

NOMBRE	RF7 - ACTUALIZACIÓN DE LA INFORMACIÓN
FUENTE	ANALISTA
DESCRIPCIÓN	La información contenida en la aplicación y la generada por la herramienta deberá de estar continuamente sometida a procesos de actualización (inserción de información nueva y modificación de información existente).
PRIORIDAD	ALTA/NECESARIO

NOMBRE	RF8 - MANTENIBLE- SOSTENIBLE
FUENTE	ANALISTA
DESCRIPCIÓN	El software diseñado debe de ser mantenible y sostenible durante todo su tiempo de vida.
PRIORIDAD	ALTA/NECESARIO

NOMBRE	RF9 - RECUPERABLE
FUENTE	ANALISTA
DESCRIPCIÓN	La información contenida en la aplicación deberá de estar sujeta a copias de seguridad, además debe de realizarse un backup semanalmente para recuperar toda la información disponible una vez que el sistema caiga por cualquier fallo.
PRIORIDAD	ALTA/NECESARIO

NOMBRE	RF10 - EXTENSIBLE A TODOS LOS USUARIOS
FUENTE	ANALISTA
DESCRIPCIÓN	La aplicación web debe de estar disponible y accesible por parte de todos los usuarios al igual que se ha de garantizar que todo los usuarios de la compañía podrán usar la herramienta para las pruebas de regresión independientemente del host en el que trabajen.
PRIORIDAD	ALTA/NECESARIO

2. Requisitos de rendimiento

NOMBRE	RRA 1 - TIEMPO DE RESPUESTA
FUENTE	ANALISTA
DESCRIPCIÓN	Tanto la aplicación como la herramienta de test deben de responder a la petición que realice el usuario en un tiempo determinado.
PRIORIDAD	ALTA/NECESARIO

3. Requisitos tecnológicos

NOMBRE	RT 1 - MULTIPLATAFORMA
FUENTE	ANALISTA
DESCRIPCIÓN	La aplicación web debe de estar disponible para cualquier S.O. alojado en los equipos y dispositivos.
PRIORIDAD	ALTA/NECESARIO

NOMBRE	RT 2 - EXPORTABLE Y REUTILIZABLE
FUENTE	ANALISTA
DESCRIPCIÓN	El software de la aplicación Ty de la herramienta tendrán un carácter reutilizable y exportable el cual permitirá realizar nuevas versiones con mejoras incorporadas a partir de los originales.
PRIORIDAD	ALTA/NECESARIO

NOMBRE	RT 3 - ALMACENAMIENTO FLEXIBLE
FUENTE	ANALISTA
DESCRIPCIÓN	La aplicación web debe de soportar cambios en las bases de datos sin que la aplicación deje de funcionar correctamente, siempre y cuando no se toquen las tablas correspondientes a las ya existentes.
PRIORIDAD	ALTA/NECESARIO

4. Atributos del sistema

1. Requisitos de seguridad

NOMBRE	RS1 - ACTIVIDAD DEL USUARIO
FUENTE	ANALISTA
DESCRIPCIÓN	<p>La aplicación web debe de estar en todo momento dentro del marco fijado por las instrucciones del usuario, siempre y cuando este realice un comportamiento adecuado de la aplicación.</p> <p>Además con respecto a la herramienta, los usuarios se han de limitar exclusivamente a insertar el conjunto de ficheros cpp modificados puesto que la herramienta únicamente con esta información devolverá los sets de pruebas de regresión a ejecutar para el testeo del código.</p>
PRIORIDAD	ALTA/NECESARIO

NOMBRE	RS2 - INYECCIONES SQL
FUENTE	ANALISTA
DESCRIPCIÓN	El sistema debe de evitar ataques de inyección de sentencias SQL por parte de usuarios no deseados.
PRIORIDAD	ALTA/NECESARIO

NOMBRE	RS3 - UBICACIÓN DE LAS INTERFACES DE ADMINISTRACIÓN
FUENTE	ANALISTA
DESCRIPCIÓN	Las interfaces de administración se deben de ubicar en un servidor seguro, pudiendo usar a su vez interfaces de escritorio.
PRIORIDAD	ALTA/NECESARIO

2. Requisitos de fiabilidad y disponibilidad

NOMBRE	RD1 - DISPONIBILIDAD Y SERVICIO
FUENTE	ANALISTA
DESCRIPCIÓN	La aplicación debe de albergar un sistema el cual responda siempre, es decir, debe de estar disponible y en servicio en cualquier momento. Al igual debe de ocurrir con la herramienta, debe de estar siempre disponible para su uso.
PRIORIDAD	MEDIO/DESEADO

NOMBRE	RD2 - BACKUP
FUENTE	ANALISTA
DESCRIPCIÓN	El sistema en cualquier momento, en caso de fallo o cualquier otro problema, debe de recuperarse en un período razonable (24 horas) desde el momento en el que fue detectado el fallo.
PRIORIDAD	MEDIO/DESEADO

3. Requisitos de implementación

NOMBRE	RI 1 - MANTENIBLE
FUENTE	ANALISTA
DESCRIPCIÓN	Tanto la aplicación como la herramienta de tests deben de ser fácilmente mantenibles.
PRIORIDAD	ALTA/NECESARIO

NOMBRE	RI 2 - ALMACENAMIENTO DE LOS USUARIOS
FUENTE	ANALISTA
DESCRIPCIÓN	El sistema de la aplicación debe de albergar espacio suficiente para aumentar los registros de los usuarios.
PRIORIDAD	MEDIA/DESEADO

NOMBRE	RI 2 - ALMACENAMIENTO DE LOS SUBSETS DE REGRESIONES CON SUS DISTINTOS ESCENARIOS (MODULO 1)
FUENTE	ANALISTA
DESCRIPCIÓN	El módulo de almacenamiento de subsets de regresiones debe de contar con una base de datos la cual debe de tener espacio suficiente para aumentar sus contenidos.
PRIORIDAD	MEDIA/DESEADO

NOMBRE	RI 2 - ALMACENAMIENTO DE LOS NOMBRES DE LAS FUNCIONES (MODULO 2)
FUENTE	ANALISTA
DESCRIPCIÓN	El Módulo 2 de la herramienta debe de albergar espacio suficiente para aumentar los contenidos correspondientes a los nombres de funciones contenidos en los ficheros de código c++.
PRIORIDAD	MEDIA/DESEADO

5. Otros requisitos

1. Requisitos económicos

NOMBRE	RE 1 - CUANTÍA ECONÓMICA
FUENTE	ANALISTA
DESCRIPCIÓN	No se tiene ninguna restricción en cuanto a costes económicos.
PRIORIDAD	MEDIA/DESEADA

NOMBRE	RE 2 - SOFTWARE
FUENTE	ANALISTA
DESCRIPCIÓN	Todo el software para la realización de este proyecto queda en el marco de la empresa.
PRIORIDAD	MEDIA/DESEADO

2. Requisitos de accesibilidad

NOMBRE	RA 1 - ACCESO A LA APLICACIÓN
FUENTE	ANALISTA
DESCRIPCIÓN	Nuestra aplicación debe de soportar todos los accesos concurrentes que se hagan y responder con un funcionamiento correcto.
PRIORIDAD	ALTA/NECESARIO

3. Requisitos futuros

NOMBRE	RFUT 1 – PREFERENCIAS DE USUARIO
FUENTE	ANALISTA
DESCRIPCIÓN	Tanto la aplicación como la herramienta de test deben de responder a la petición que realice el usuario en un tiempo determinado.
PRIORIDAD	BAJA

Catalogación de requisitos

A continuación en la siguiente tabla procedemos a realizar un pequeño catálogo de los requisitos obtenidos:

Catálogo de requisitos según la viabilidad	Tipos
Requisitos funcionales (RF) a) Prioridad alta/necesario b) Prioridad media/deseado c) Prioridad baja/opcional	Requisitos Funcionales Requisitos de Rendimiento Requisitos Tecnológicos Requisitos de Seguridad
Requisitos no funcionales (RNF) d) Prioridad alta/necesario e) Prioridad media/deseado f) Prioridad baja/opcional	Requisitos de Fiabilidad y disponibilidad Requisitos de Implementación Requisitos Económicos Requisitos de Accesibilidad

a) Requisitos con prioridad alta/necesario

Referencia	Nombre
RF1	Usuarios de la aplicación web
RF2	Usuario administrador de la aplicación web
RF3	Usuario de consulta de la aplicación web
RF4	Interfaz en inglés
RF5	Simplificación de la interacción humana
RF6	Interfaces elementales
RF7	Actualización de la información
RF8	Mantenible/sostenible
RF9	Recuperable
RF10	Extensible a todos los usuarios
RRA1	Tiempo de respuesta
RT1	Multiplataforma
RT2	Exportable y reutilizable
RT3	Almacenamiento flexible
RS2	Inyecciones SQL
RS3	Ubicación de las interfaces de administración
RI1	Mantenible
RAA1	Acceso a la aplicación

b) Requisitos con prioridad media/deseado

Referencia	Nombre
RS1	Actividad del Usuario
RD1	Disponibilidad y servicio
RD2	Backup
RI2	Almacenamiento de los usuarios
RI3	Almacenamiento de los subset de regresión del módulo 1
RI4	Almacenamiento de los nombres de funciones del módulo 2
RE1	Cuantía económica
RE2	Software

c) Requisitos con prioridad baja/opcional

Referencia	Nombre
RFUT1	Preferencias de usuario

3.5. Análisis y diagramas

3.5.1. Actores

La aplicación va a caracterizarse por tener una interacción principalmente de dos actores representados en las figuras del usuario administrador y de carga localizados en el actor “administrador” y el usuario de consulta representado por el actor “usuario”.

A continuación en las siguientes tablas se mostrarán las características y las funcionalidades de cada uno de los actores del sistema siguiendo las pautas de la siguiente plantilla.

❖ Plantilla del actor Administrador

Actor	Administrador	Ac-1
Descripción	Representa la persona que realiza la gestión de la aplicación	
Características	Introduce los datos correspondientes a cada uno de los nuevos usuarios de pruebas de regresión, así como actualizar sus datos, consultarlos, migrarlos o simplemente eliminarlos.	
Relaciones		
Referencias		
Autor	M.Victoria Santiago Alcalá	Fecha 20/04/2018 Versión 1.1

Atributos		
Nombre	Descripción	Tipo

Comentarios

❖ **Plantilla del actor Usuario**

Actor	Usuario	Ac-2		
Descripción	En esta figura podemos ver representada a cualquier persona del equipo que necesite bien consultar la información de los usuarios de regresión o de los puertos o bases de datos, este usuario también es el mismo usuario que usara la herramienta para conocer los sets de pruebas de regresión que ha de ejecutar para testear el proyecto.			
Características	<p>Tiene acceso a todos los datos relacionados con los puertos, bases de datos usadas y alias.</p> <p>Ademas tiene acceso a la ejecución en el entorno que le corresponda de las pruebas de regresión.</p>			
Relaciones				
Referencias				
Autor	M.Victoria Santiago Alcalá	Fecha 20/04/2018	Versión 1.1	

1. Atributos		
Nombre	1. Descripción	Tipo

Comentarios

3.5.2. Diagramas de casos de uso

Descriptas ya las principales funcionalidades de cada uno de los actores que intervienen en el sistema, procedemos a crear los diagramas de uso correspondientes a las funcionalidades que pueden ser llevadas a cabo por cada uno de los actores.

A continuación muestro el diagrama de uso correspondiente a las acciones que puede realizar el actor Administrador (Ac-1):

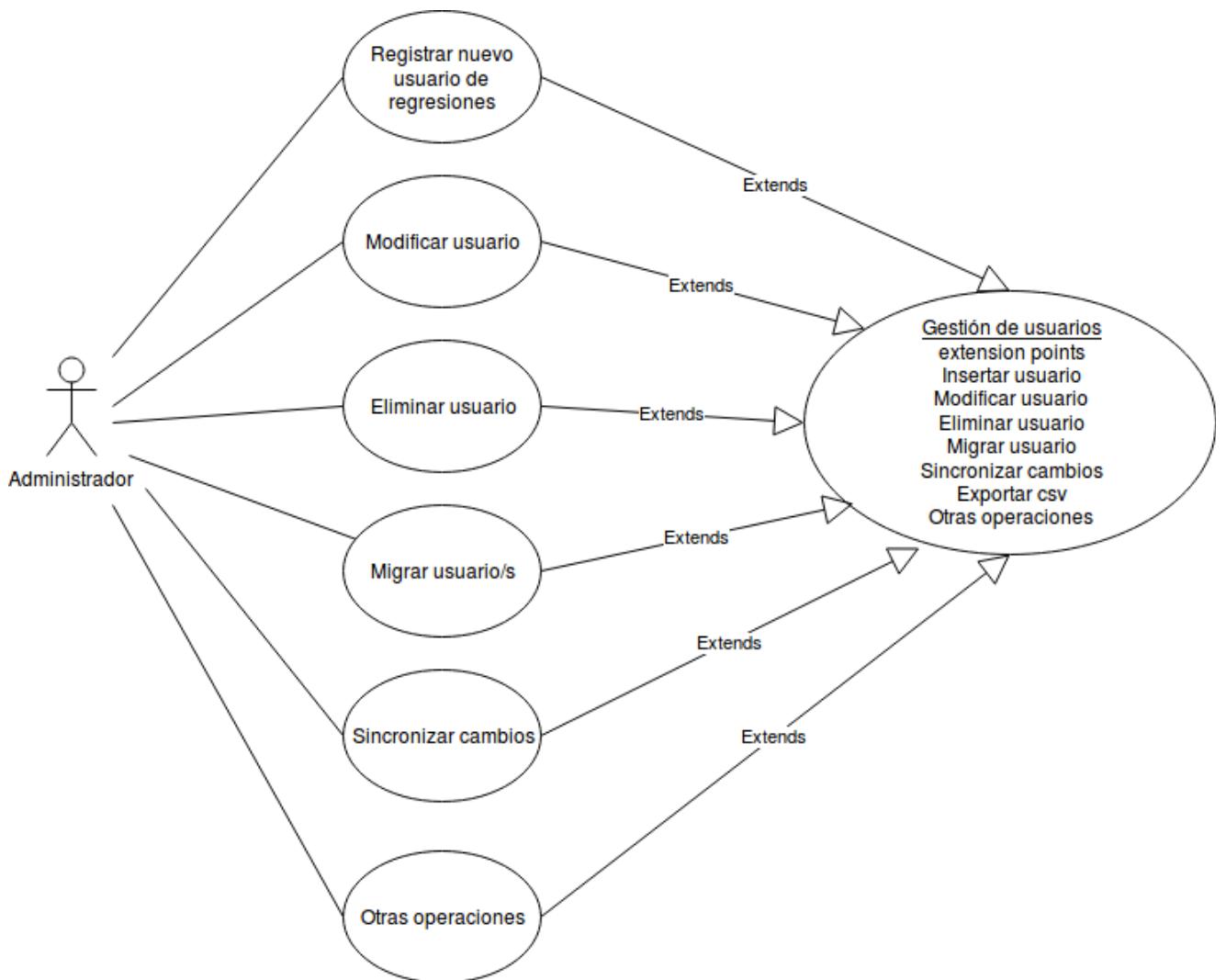


Diagrama 2. Caso de uso del actor Administrador.

Caso de Uso	Incluir/Excluir usuario de regresión	CU-2
Actores	Administrador	

Propósito
Permitir al administrador manejar toda la información sobre un usuario.

Curso Normal			
1	El administrador accede al sistema.		
		2	El sistema le da la opción de insertar nuevo usuario o modificar o eliminar un usuario ya existente en el sistema.
3	El administrador elige opción y le da todos los datos correspondiente a la opción elegida al sistema.		
		4	El sistema recibe toda la información y realiza la acción que el administrador le ha pedido que realice.
5	Cuando el administrador termina con un usuario, puede o bien seguir con otro usuario, realizar búsquedas o salir de la aplicación.		
		6	El sistema seguirá obedeciendo al administrador, en lo que a las instrucciones de este se refiere.

Diagrama de uso del actor Usuario (Ac-2):

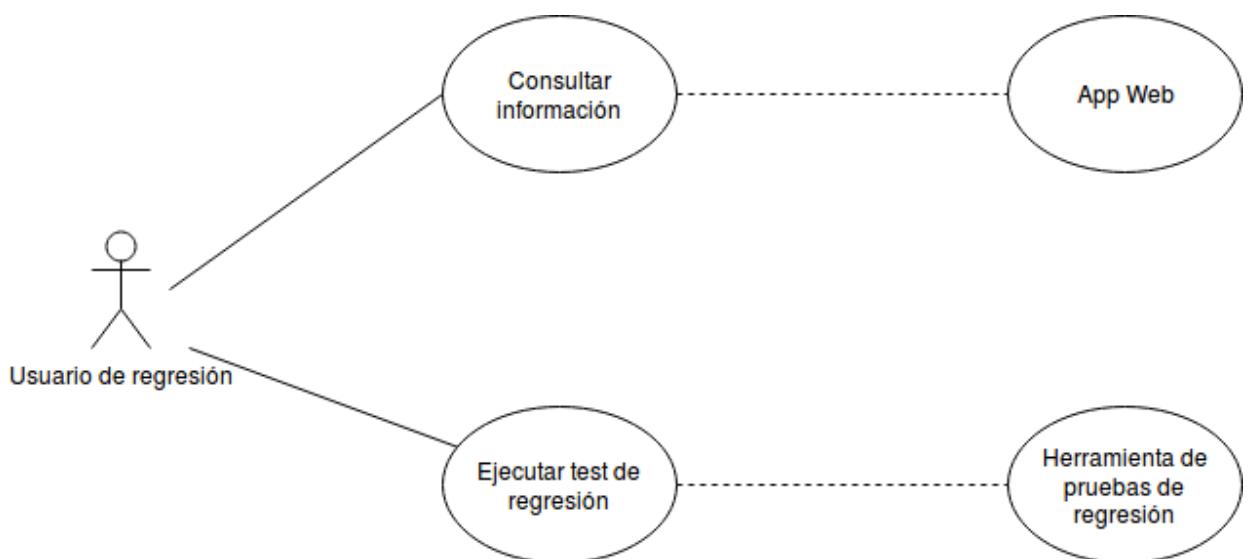


Diagrama 3. Diagrama de uso del actor Usuario AC-2

y su descripción utilizando una plantilla:

Caso de Uso	Consultar usuario/puerto/base de datos/alias y ejecutar pruebas de regresión	CU-1
Actores	Usuario	

Propósito
Permitir al usuario ver la información sobre los puertos, bases de datos, alias, etc.
Permitir al usuario ejecutar test de regresión.

Curso Normal			
1	El usuario accede al sistema.		
		2	El sistema le pide que se autentifique.
3	El usuario se autentifica y bien realiza la consulta de información o realiza la petición para ejecutar el test de regresión.		
		4	El sistema le da toda la información correspondiente asociada a la solicitud realizada o le permite acceder al entorno de ejecución de pruebas de regresión.
5	Cuando el usuario ha terminado, puede o bien realiza otra consulta , ejecuta otro test de regresión o bien abandona la aplicación/herramienta.		
		6	El sistema seguirá obedeciendo al usuario, en lo que a las instrucciones de este se refiere.

3.5.3. Diagramas de actividad

A priori, antes de poder realizar los diagramas de actividad, se necesita crear las plantillas de los casos de uso con información más extendida para después asignarle a cada diagrama de actividad una de estas plantillas de caso de uso.

Casos de uso de la aplicación web:

Caso de Uso	Consultar usuario			CU-1
Actores	Usuario			
Tipo	Primario, Esencial			
Referencias	CU-2, CU-3			
Precondición				
Postcondición	Aparece toda la información correspondiente a un usuario de regresión.			
Autor	M.Victoria Santiago Alcalá	Fecha	22/05/2018	Versión 1. 1

Propósito
Visualizar toda la información sobre un usuario.

Resumen
El usuario se registra en el sistema, seguidamente realiza una consulta, finalmente obtiene la información requerida.

Curso Normal				
1	Usuario: entra en la aplicación web.	2	Espera hasta que el usuario le proporcione una consulta.	
3	Usuario: realiza la consulta	4	Recibe la consulta y le proporciona al usuario la información correspondiente a dicha consulta.	
5	Usuario: visualiza la información.			
6	Usuario: cierra la aplicación.			

Otros datos			
Frecuencia esperada		Rendimiento	
Importancia		Urgencia	
Estado		Estabilidad	

Comentarios

Caso de Uso	Información			CU-2
Actores	Usuario			
Tipo	Primario, Esencial			
Referencias				
Precondición				
Postcondición	Proporciona toda la información correspondiente a un usuario, puerto, base de datos, entorno.			
Autor	M.Victoria Alcalá	Santiago	Fecha	22/05/2018
			Versión	1.1

Propósito
Proporcionar al usuario toda la información correspondiente a su consulta, ya sea información referente a los puertos usados, a los entornos, etc.

Resumen

El sistema recibe una solicitud a cerca de una consulta, devuelve la información disponible correspondiente a dicha consulta.

Curso Normal

1	Usuario: entra en la aplicación web.	2	Espera hasta que el usuario le proporcione una consulta.
3	Usuario: usuario realiza la consulta	4	Recibe la consulta y le proporciona al usuario toda la información correspondiente.
5	Usuario: decide testear el código c++ modificado con las pruebas de regresión.	6	Por medio de la herramienta se consigue las correspondientes sets de regresiones a ejecutar para testear las modificaciones y se devuelven al usuario.
7	Usuario: Ejecuta los test de regresión.	8	Devuelve el resultado de los test.

Otros datos

Frecuencia esperada		Rendimiento	
Importancia		Urgencia	
Estado		Estabilidad	

Comentarios

Caso de Uso	Insertar Usuario			CU-3
Actores	Administrador			
Tipo	Primario, Esencial			
Referencias				
Precondición				
Postcondición	Se añade una nuevo usuario.			
Autor	M.Victoria Santiago Alcalá	Fecha	22/05/2018	Versión 1.1

Propósito
Añadir una nuevo usuario para poder ejecutar test de regresión.

Resumen
El administrador decide insertar un nuevo usuario por medio de la dashboard web de Django la cual va a permitir que al registrar el usuario, pueda poder testear su código de forma simple por medio de los test de regresión.

Curso Normal			
1	Administrador: entra en la aplicación.	2	Espera hasta que el administrador le proporcione instrucciones.
3	Administrador: decide insertar un nuevo usuario.	4	Le proporciona al administrador todos los datos que son necesarios para que este cree el nuevo usuario.
5	Administrador: introduce todos los datos del usuario.	6	Recibe todos los datos.
		7	Guarda los nuevos datos
		8	Actualiza la aplicación para que a partir de ahora los usuarios puedan consultar el nuevo usuario y este pueda lanzar pruebas de re-

			gresión.

Otros datos			
Frecuencia esperada		Rendimiento	
Importancia		Urgencia	
Estado		Estabilidad	

Comentarios

Caso de Uso	Modificar Usuario			CU-4
Actores	Administrador			
Tipo	Primario, Esencial			
Referencias				
Precondición				
Postcondición	Se modifican los datos del usuario.			
Autor	M.Victoria Santiago Alcalá	Fecha	22/05/2018	Versión 1.1

Propósito
Modificar los datos de un usuario

Resumen
El administrador decide modificar los datos correspondientes a un usuario, bien porque estén equivocados o porque desea añadir más información.

Curso Normal			
1	Administrador: entra en la aplicación.	2	Espera hasta que el administrador le proporcione instrucciones.
3	Administrador: decide modificar un usuario.	4	Le proporciona al administrador todos los usuarios disponibles en el sistema
5	Administrador: elige el usuario del cual desea modificar su información.	6	Devuelve al administrador todos los datos correspondientes al usuario seleccionado.
7	Administrador: modifica todos los datos del usuario.	8	Recibe los nuevos datos para el usuario.
		9	Guarda los nuevos datos.
		10	Actualiza la aplicación para que a partir de ahora los usuarios puedan visualizar los nuevos datos modificados.

Otros datos			
Frecuencia esperada		Rendimiento	
Importancia		Urgencia	
Estado		Estabilidad	

Comentarios

Caso de Uso	Eliminar Usuario				CU-5
Actores	Administrador				
Tipo	Primario, Esencial				
Referencias					
Precondición					
Postcondición	Se elimina el usuario.				
Autor	M.Victoria Santiago Alcalá	Fecha	22/05/2018	Ver-sión	1.1

Propósito
Eliminar un usuario del sistema.

Resumen
El administrador decide eliminar un usuario por completo del sistema, bien porque no va a ejecutar mas test de regresión o porque dejó la compañía.

Curso Normal					
1	Administrador: entra en la aplicación.	2	Espera hasta que el administrador le proporcione instrucciones.		
3	Administrador: decide eliminar un usuario.	4	Le proporciona al administrador todos los usuarios disponibles en el sistema.		
5	Administrador: elige el usuario que desea eliminar.	6	Devuelve al administrador todos los datos correspondientes al usuario seleccionado.		
7	Administrador: confirma que desea eliminar por completo el usuario.	8	Recibe la orden de eliminar el usuario.		
		9	Borra todos los datos existentes sobre el usuario.		
		10	Actualiza la aplicación para que a partir de ahora los usuarios no puedan volver a visualizar ese usuario recién		

			eliminado y no pueda ejecutar mas test.
--	--	--	---

Otros datos			
Frecuencia esperada		Rendimiento	
Importancia		Urgencia	
Estado		Estabilidad	

Comentarios

Caso de Uso	Migrar Usuario			CU-6
Actores	Administrador			
Tipo	Primario, Esencial			
Referencias				
Precondición				
Postcondición	Se migra el usuario.			
Autor	M.Victoria Santiago Alcalá	Fecha	22/05/2018	Ver-sión
				1.1

Propósito
Migar un usuario.

Resumen
El administrador decide migrar un usuario del entorno o base de datos hacia otra.

Curso Normal			
1	Administrador: entra en la aplicación.	2	Espera hasta que el administrador le proporcione instrucciones.
3	Administrador: decide migrar un usuario.	4	Le proporciona al administrador todos los usuarios disponibles en el sistema.
5	Administrador: elige el usuario o usuarios que desea migrar con el checkbox y selecciona el nombre de la base de datos a la que se migra.	6	Migra los usuarios seleccionados.

Otros datos			
Frecuencia esperada		Rendimiento	
Importancia		Urgencia	
Estado		Estabilidad	

Comentarios

Caso de Uso	Exportar csv			CU-6
Actores	Administrador, usuario			
Tipo	Primario, Esencial			
Referencias				
Precondición				
Postcondición	Se exporta la lista de usuarios a un fichero csv.			
Autor	M.Victoria Santiago Alcalá	Fecha	22/05/2018	Versión 1.1

Propósito
Exportar lista de usuarios.

Resumen
El administrador o usuario quiere obtener la lista de usuarios, puertos, bases de datos, etc.

Curso Normal				
1	Administrador/Usuario: entra en la aplicación.	2	Espera hasta que el administrador le proporcione instrucciones.	
3	Administrador/Usuario: decide exportar los usuarios y realizar la petición.	4	Le proporciona al administrador todos los usuarios disponibles en el sistema y crea el fichero csv devolviéndoselo.	
5	Administrador/Usuario: obtienen el fichero csv con el listado.			

Otros datos			
Frecuencia esperada		Rendimiento	
Importancia		Urgencia	
Estado		Estabilidad	

Comentarios

Casos de uso de la herramienta de tests:

Caso de Uso	Consultar regresiones a ejecutar				CU-6
Actores	Usuario				
Tipo	Primario, Esencial				
Referencias					
Precondición					
Postcondición	Obtener subset de regresiones a ejecutar.				
Autor	M.Victoria Santiago Alcalá	Fecha	22/05/2018	Versión	1.1

Propósito

Obtener el set de regresiones que se ha de ejecutar.

Resumen

El usuario ha introducido cambios en el código fuente del producto y al integrarlo quiere realizar las pruebas de regresión en el código por lo que usa la herramienta con el fin de conocer los subsets o sets de regresiones que ha de ejecutar para el testeo.

Curso Normal

1	Usuario: inserta los enlaces al proyecto.	2	Uso el modulo 2 donde realiza el parseo del código modificado en busca de las funciones que lo contienen.
		3	Guarda las funciones y las identifica en el fichero del modulo 1 el cual contiene las relaciones entre

			las funciones y los subset de regresiones que se han de ejecutar para testearlas.
		4	Devuelve el nombre de los subsets o sets de pruebas de regresión que se han de ejecutar para testear los nuevos cambios aplicados en el proyecto.
5	Usuario: obtiene la lista de subsets o sets de regresiones que ha de ejecutar para testear su código.		

Otros datos			
Frecuencia esperada		Rendimiento	
Importancia		Urgencia	
Estado		Estabilidad	

Comentarios

Una vez se han detallado los casos de uso más importantes, se muestran a continuación los diagramas de actividad:

- Herramienta de tests:

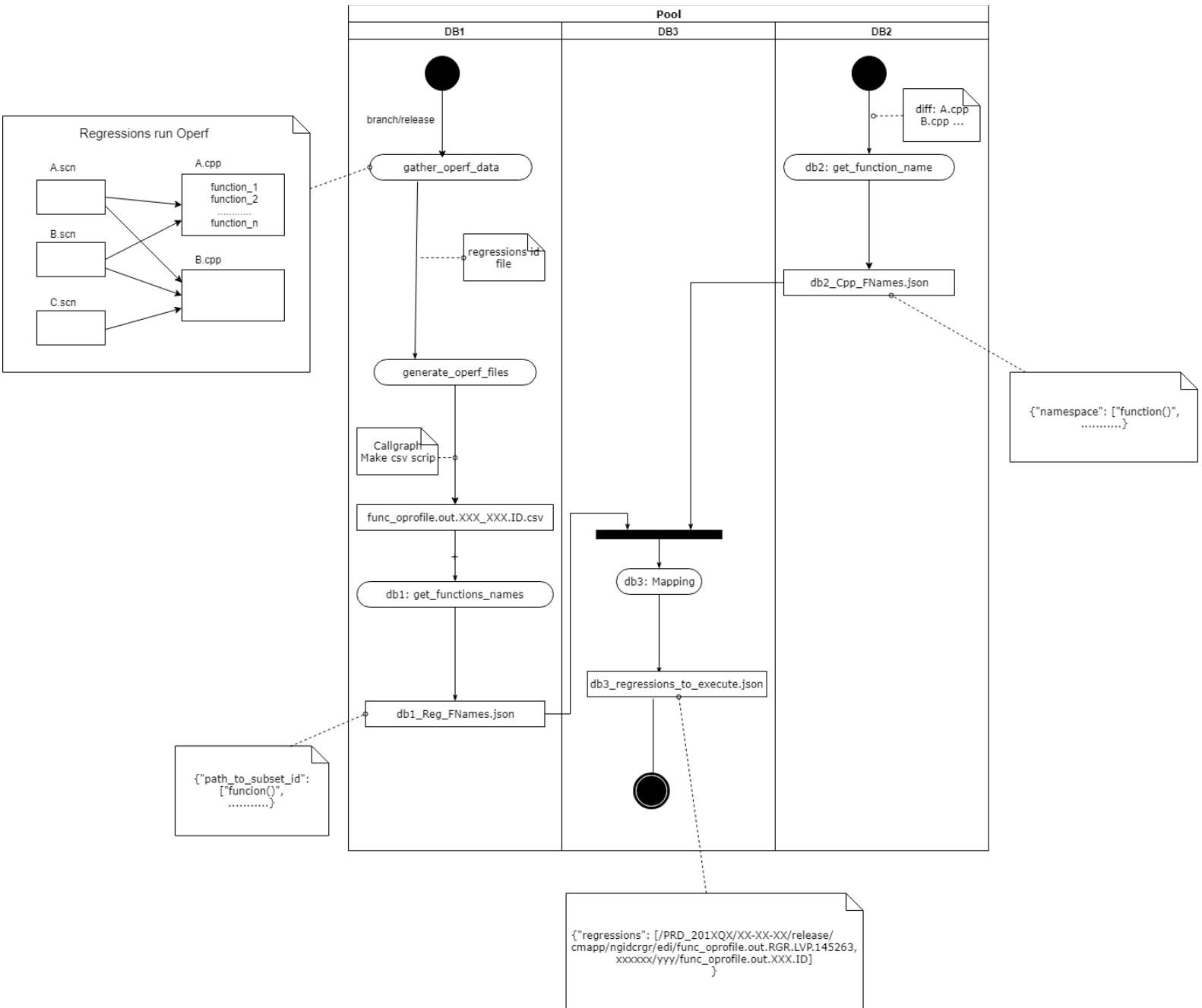


Ilustración 20. Herramienta de tests

Aplicación web para la creación de nuevos usuarios de regresión:

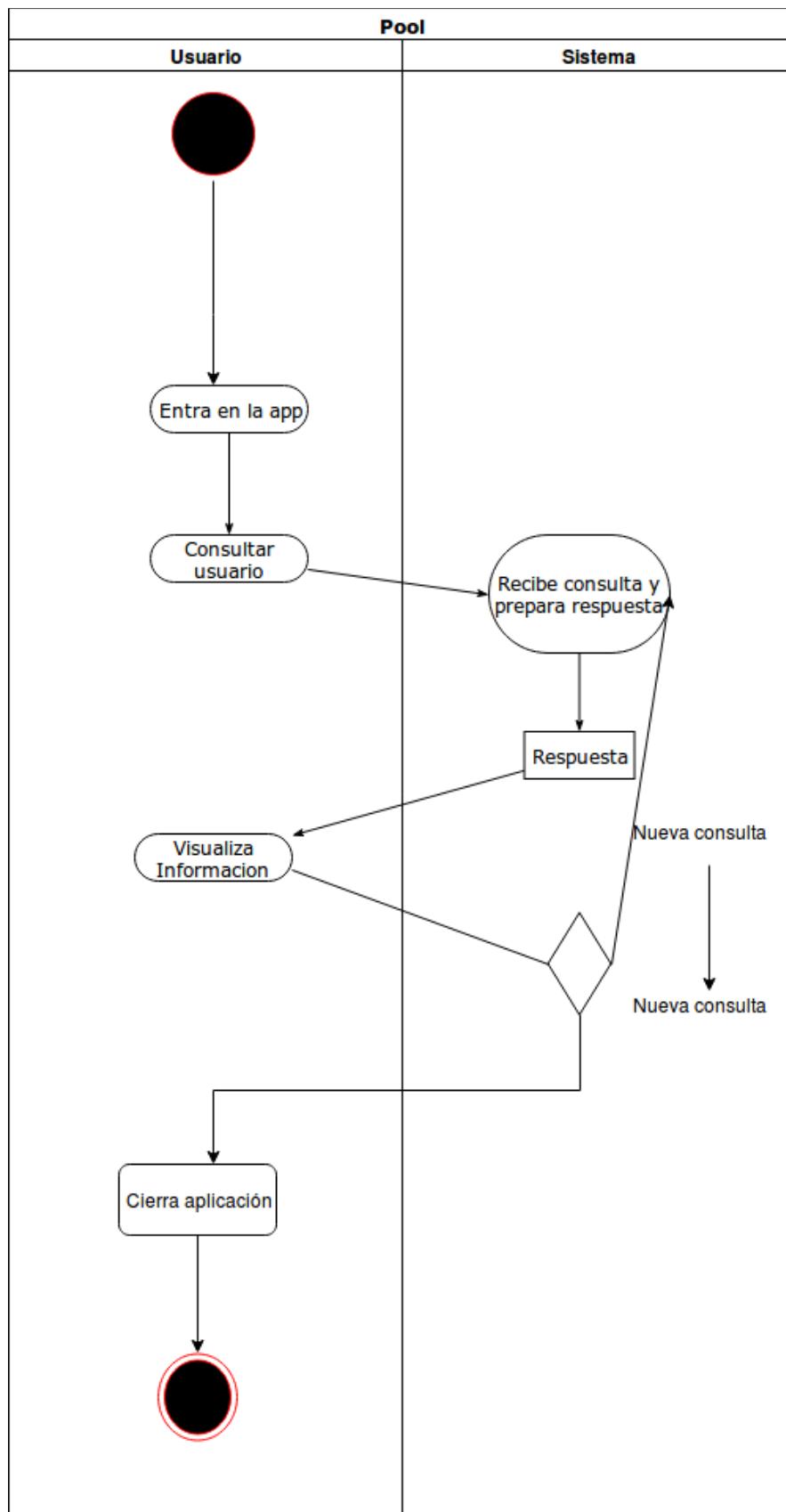


Ilustración 21. Consulta información.

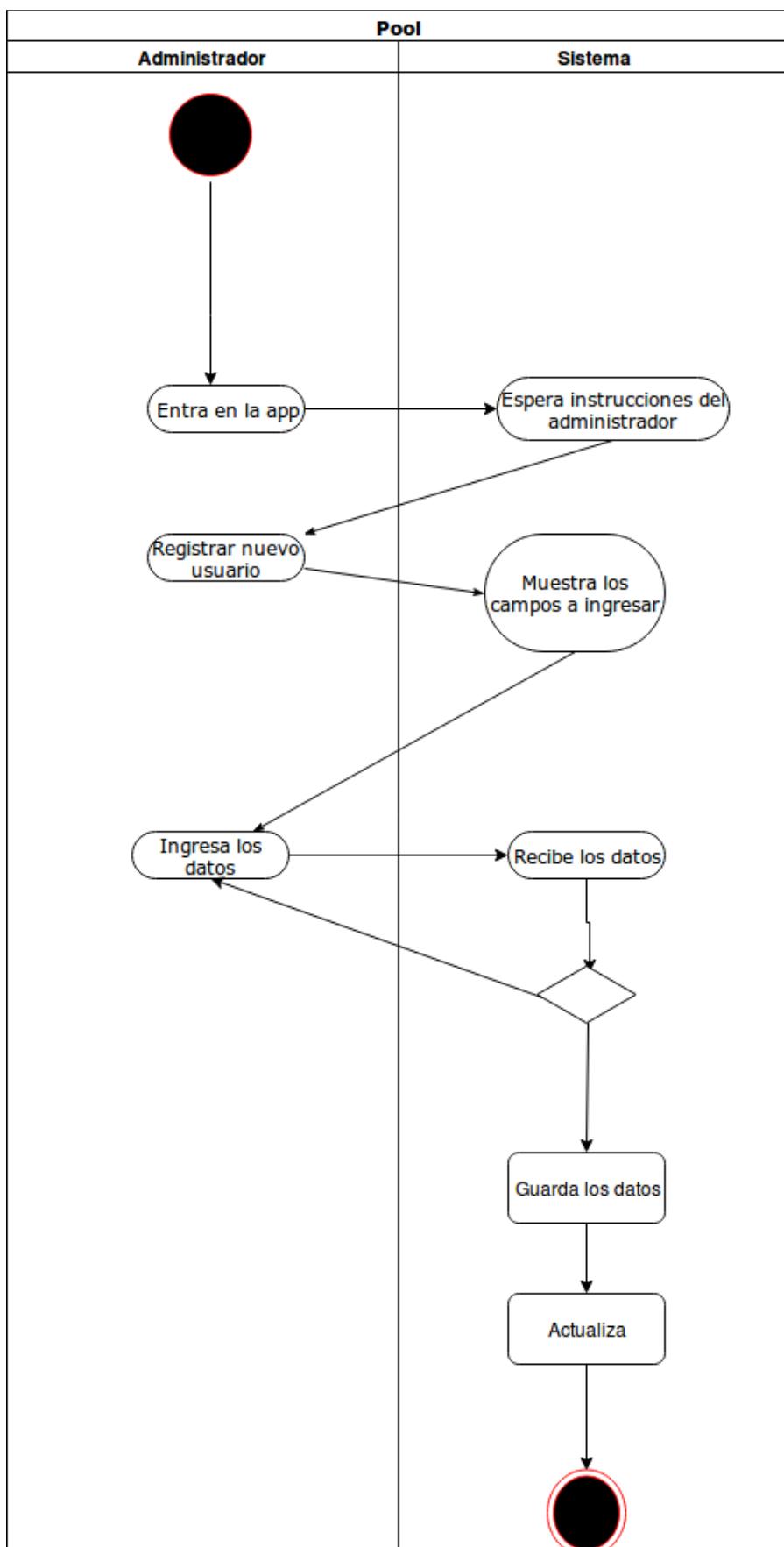


Ilustración 22. Crear nuevos usuarios

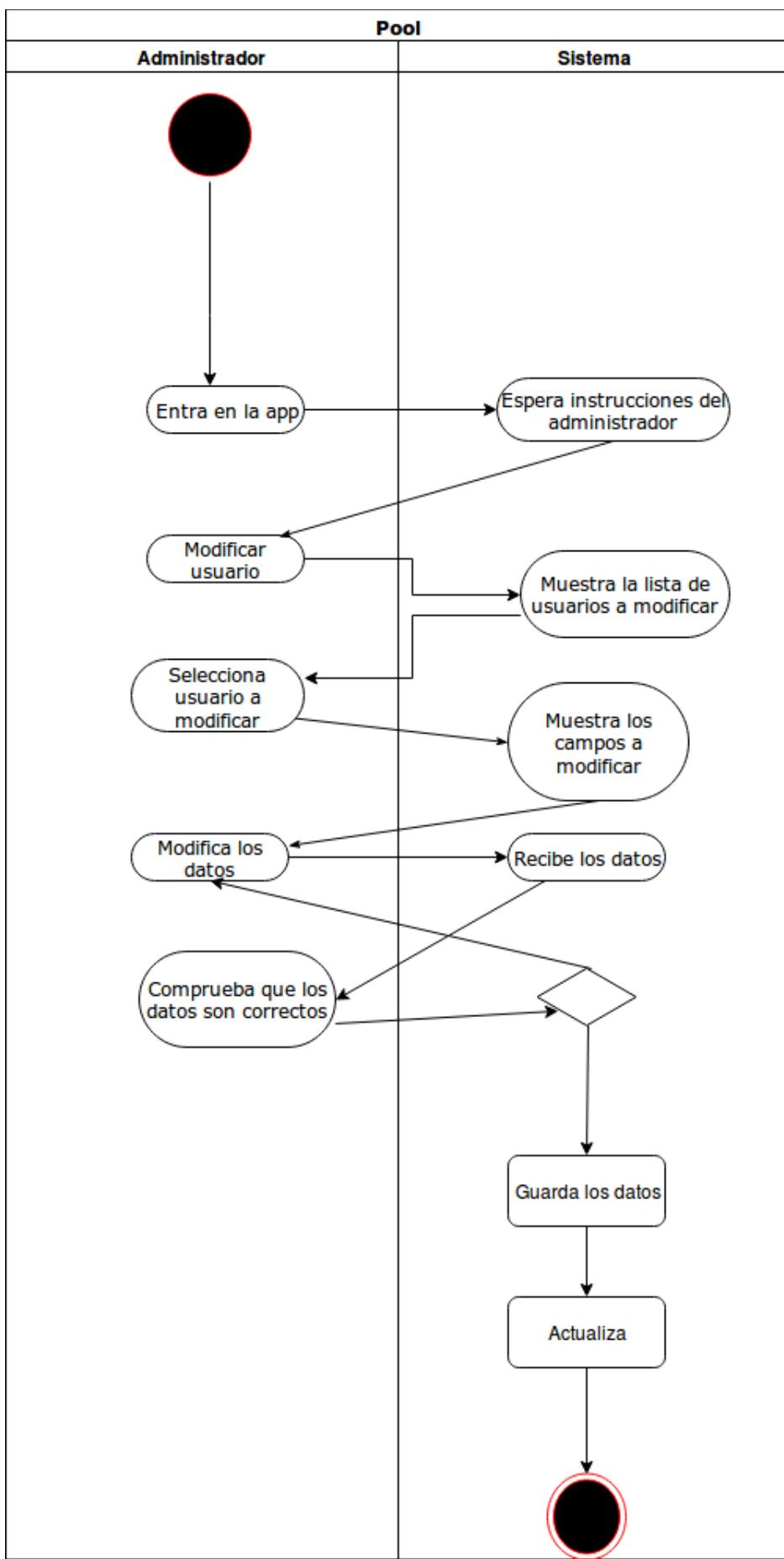


Ilustración 23. Modificar usuarios

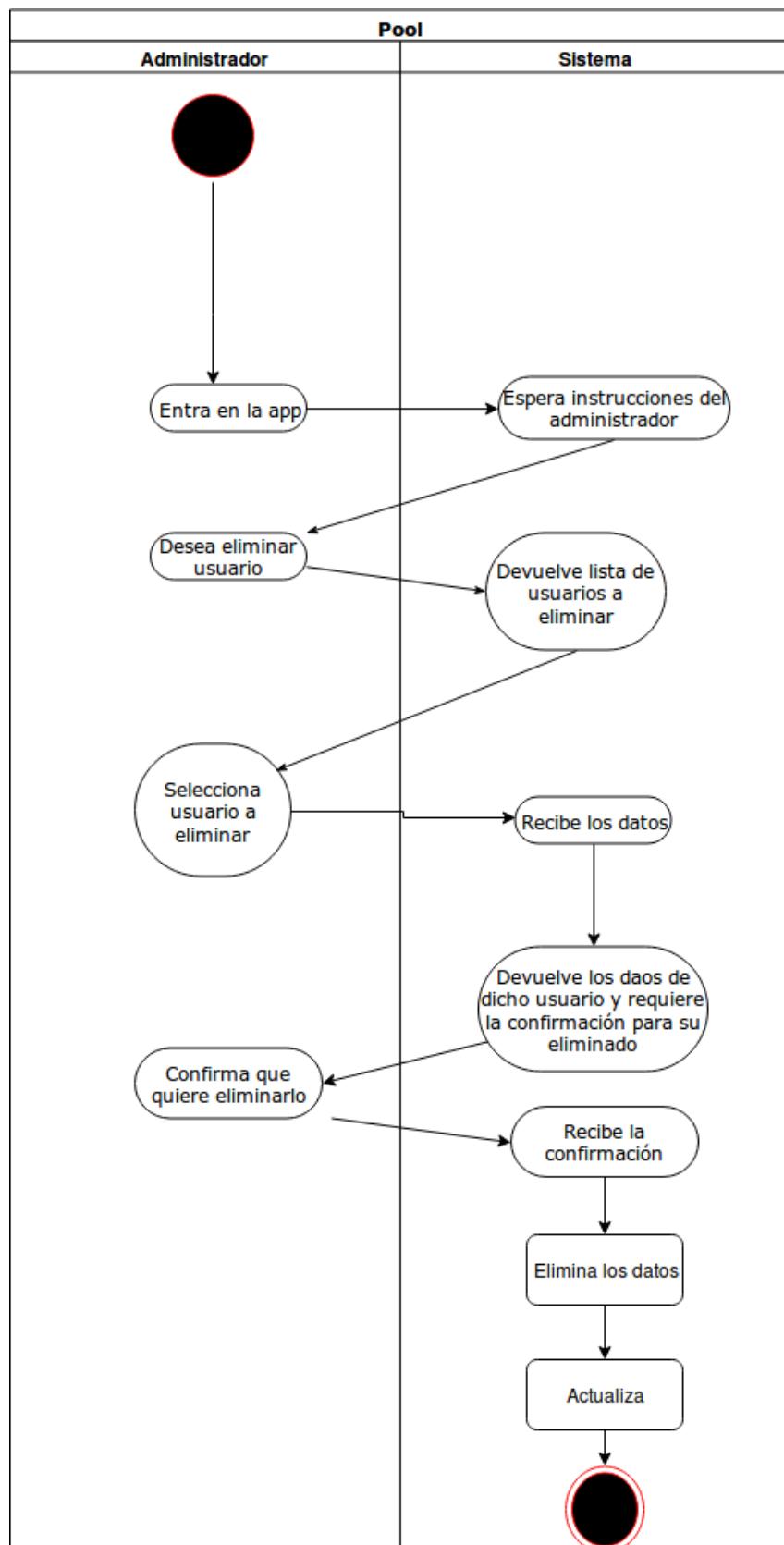


Ilustración 24. Eliminar usuario

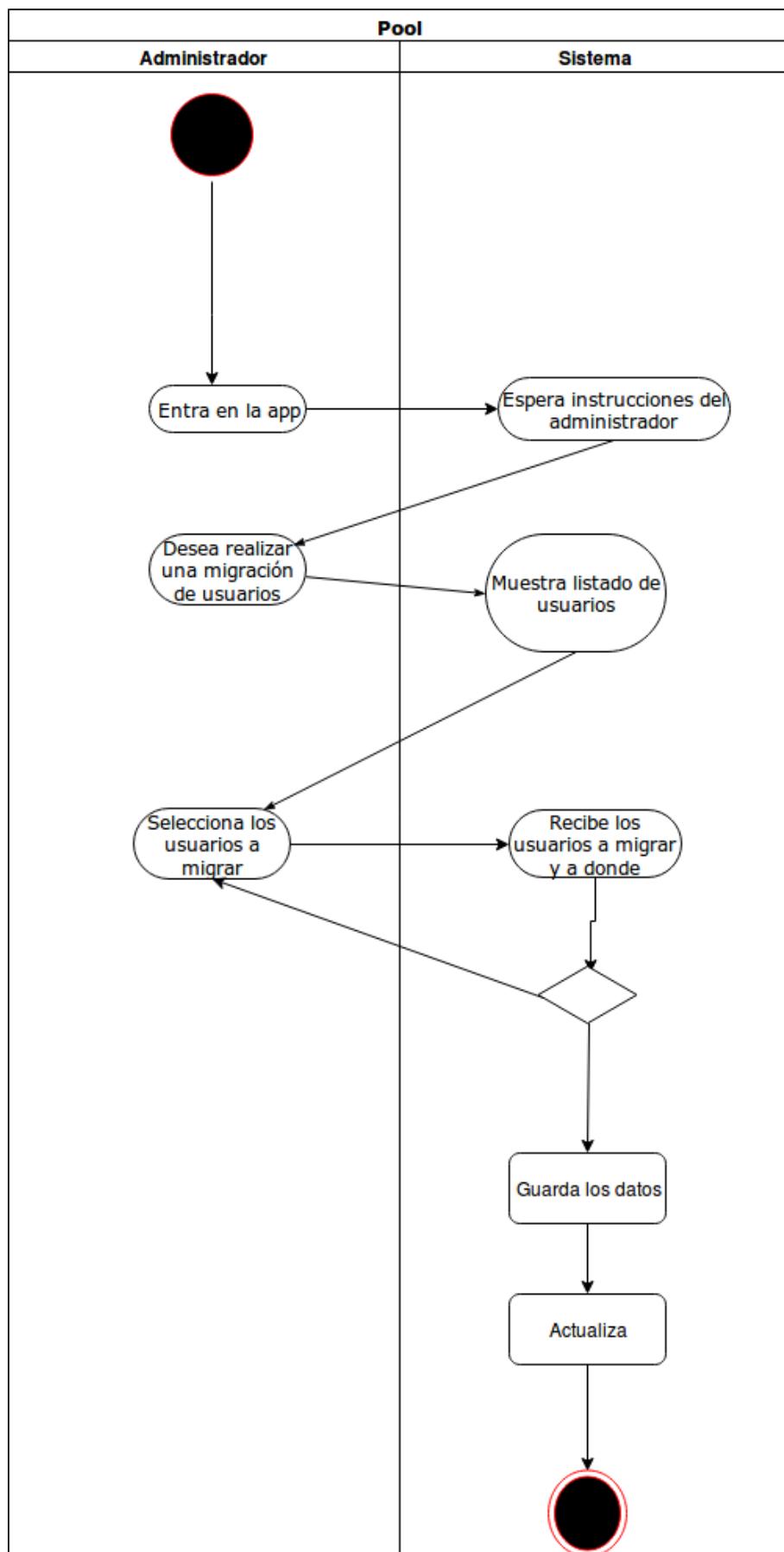


Ilustración 25. Migrar usuario

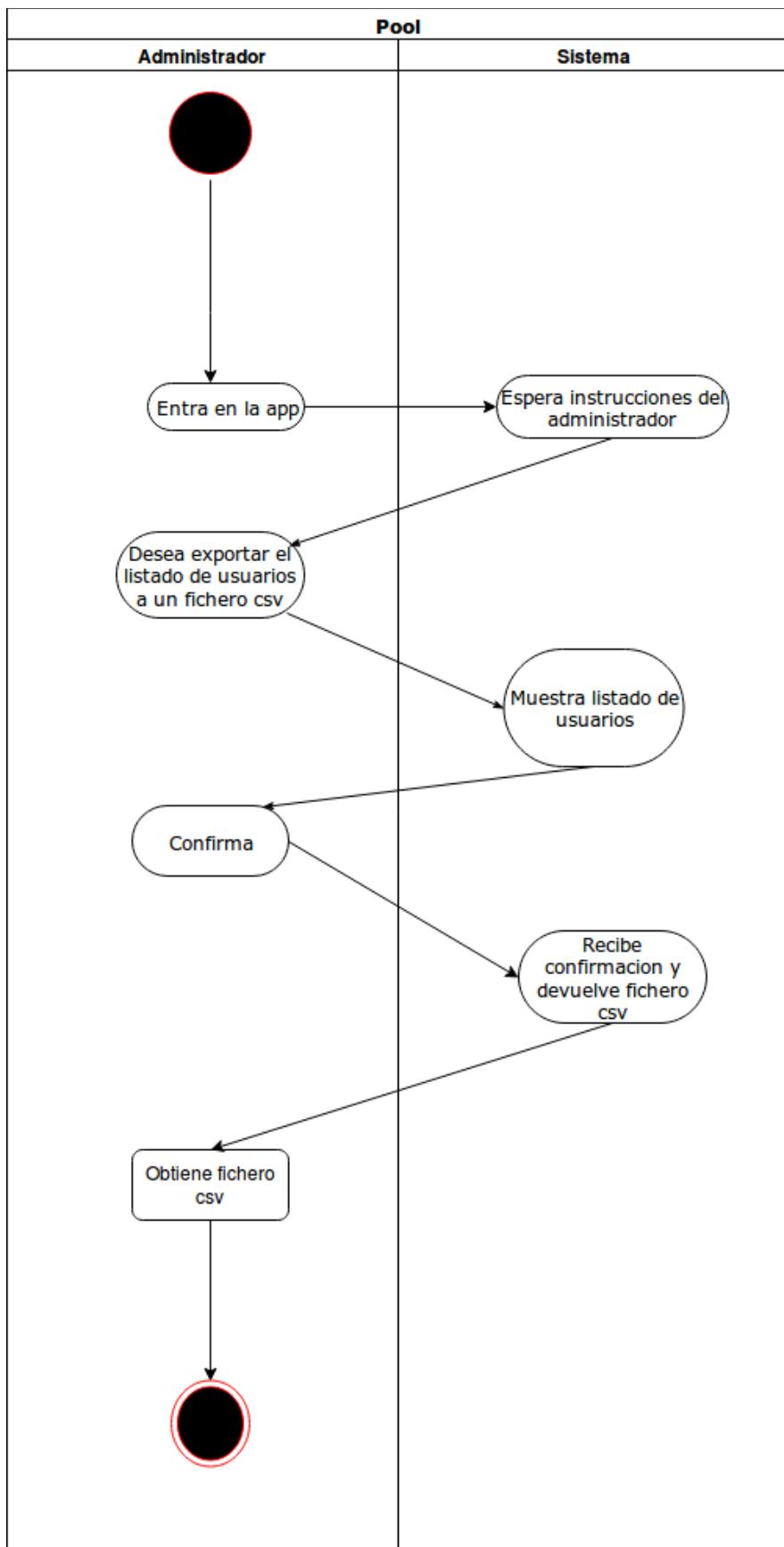


Ilustración 26. Exportar a fichero csv.

4. Diseño y alternativas

4.1. Diagrama de clases

El diagrama de clases correspondiente a la aplicación web es el siguiente:

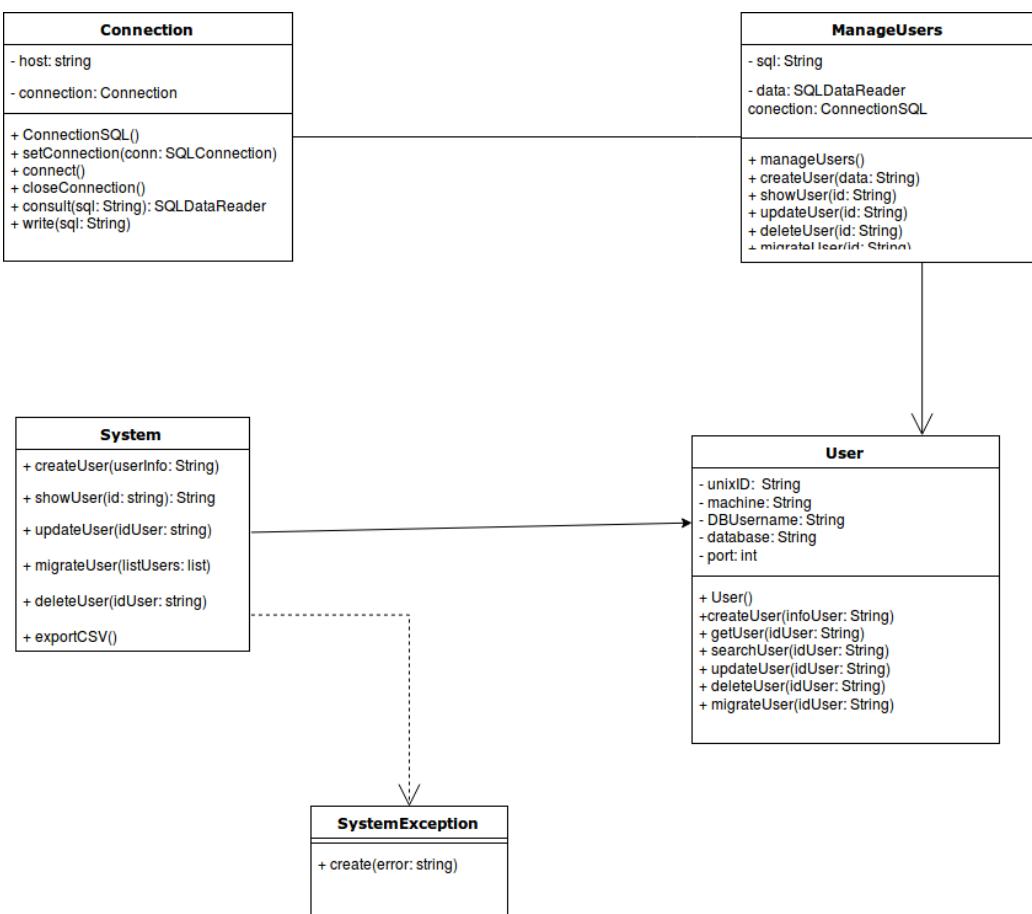


Ilustración 27. Diagrama de clases de la aplicación web

4.2. Contratos

A continuación, tras haber estudiado los diagramas anteriores tanto de la aplicación web como de la herramienta, se van a mostrar en el siguiente apartado el conjunto de contratos los cuales se hayan asociados a cada una de las funciones que ofrecen ambos.

Nombre	crearUsuario(datosUsuario)
Responsabilidades	Dar de alta un nuevo usuario.
Tipo	Sistema.
Referencias cruzadas	Caso de uso “Insertar Usuario”.
Notas	
Excepciones	El usuario ya se encuentra registrado en el sistema. Los datos del administrador son incorrectos.
Salida	
Precondición	
Postcondición	Se crea un nuevo usuario de regresión.

Nombre	modificarUsuario(idUsuario)
Responsabilidades	Modificar los datos de un usuario existente.
Tipo	Sistema.
Referencias cruzadas	Caso de uso “Modificar Usuario”.
Notas	
Excepciones	El usuario no existe en la aplicación. Los datos del administrador son incorrectos.
Salida	
Precondición	
Postcondición	Se modifica un usuario con datos nuevos de tal forma que se actualiza su información.

Nombre	eliminarUsuario(idUsuario)
Responsabilidades	Tiene la función de eliminar un usuario de regresión.
Tipo	Sistema.
Referencias cruzadas	Caso de uso “Eliminar Usuario”.
Notas	
Excepciones	El usuario no existe en la aplicación.
Salida	
Precondición	
Postcondición	Se elimina un usuario ya existente.

Nombre	migrarUsuario(list)
Responsabilidades	Tiene la función de migrar uno o varios usuarios de regresión.
Tipo	Sistema.
Referencias cruzadas	Caso de uso “Migrar Usuario”.
Notas	
Excepciones	El usuario/s no existen en la aplicación.
Salida	
Precondición	
Postcondición	Se elimina un usuario ya existente.

Nombre	exportarCSV()
Responsabilidades	Tiene la función de exportar la lista de todos los usuarios a un fichero csv.
Tipo	Sistema.
Referencias cruzadas	Caso de uso “Exportar CSV”.
Notas	
Excepciones	No hay usuarios que exportar.
Salida	
Precondición	
Postcondición	Se crea un fichero csv con los usuarios.

Nombre	datosUsuario = consultarUsuario(id)
Responsabilidades	Su responsabilidad es devolver todos los datos que se correspondan a los usuarios.
Tipo	Sistema.
Referencias cruzadas	Caso de uso “Consultar Usuario”.
Notas	
Excepciones	El id buscado no se encuentra.
Salida	
Precondición	El id debe de existir.
Postcondición	

4.3. Estudio de alternativas

En este apartado se procede a la presentación de las distintas alternativas que podrían ser candidatas para el desarrollo del proyecto. Se van a ir explotando de forma dividida según los distintos módulos de los que disponemos.

Se ha de mencionar también que no se va a incluir en esta parte la aplicación web ya que para su desarrollo no hubo alternativa ya que como requisito contaba que se desarrollase en la tecnología que usan las demás dashboards de la compañía (Python, Django, MySQL, etc).

Modulo 1. Regresiones - Escenarios

En este módulo el principal cometido consiste en generar un fichero csv en el cual se almacene la información referente a las funciones que son afectadas por cada subset de regresiones.

Inicialmente se ha de generar por cada paquete un subset, para ello, se viene ejecutando operf y callgraph con el fin de obtener estos pequeños sets.

En este módulo se necesita reunir la información necesaria sobre la conducta de nuestras regresiones durante la ejecución de las mismas, este proceso es conocido comúnmente como perfilado.

Con ello lo que se pretende es determinar las áreas de nuestro programa que pueden ser optimizadas con el fin de aumentar la velocidad de este y reducir el consumo de uso de memoria entre otros.

La finalidad de este código de “perfil” es encontrar cuellos de botella de rendimiento.

En este módulo se han estudiado dos herramientas posibles: valgrind y oprofile.

Gprof

Gprof [8] es un generador de perfiles GNU el cual usa un enfoque compuesto entre instrumentación y muestreo asistido llevados a cabo por el compilador. La instrumentación se usa para recopilar en si toda aquella información en relación con las llamadas de funciones.

Para recopilar esta información de perfil en tiempo de ejecución, se ha de usar lo que vendría a ser un proceso de muestreo.

Este proceso de muestreo quiere decir que el contador del programa se prueba a intervalos regulares interrumpiendo el programa con las interrupciones del sistema operativo. Los datos de perfil resultantes no son exactos ya que este es un proceso estadístico por lo que son más bien una aproximación.

Los pasos para crear un perfil CPU de nuestro proyecto son los siguientes:

- Inicialmente se ha de compilar y enlazar el programa con un compilador y perfilador como por ejemplo puede ser gcc -pg.
- Se ha de ejecutar el programa para generar el archivo de datos del perfil quedando este como algo parecido a “abcd.out”.
- Se ha de ejecutar finalmente gprof con el fin de analizar los datos de creación de perfiles.

Se ha de notar que siempre se ha de reconstruir la aplicación cada vez que se quieran generar los perfiles al igual que también causa una sobrecarga alta la cual es causada principalmente por la instrumentación.

Otro problema que tiene Gprof es que no admite el perfilado de aplicaciones de subprocessos múltiples y tampoco puede crear perfiles de bibliotecas compartidas. Incluso si existen soluciones temporales para obtener soporte threading3, el hecho de que no pueda realizar un perfil de las llamadas en bibliotecas compartidas, lo hace totalmente inadecuado para los proyectos actuales.

Valgrind

Es una herramienta para el análisis de ejecución en profundidad la cual funciona simulando un procesador virtual. Se caracteriza por ofrecer una serie de herramientas las cuales toman parte en el rendimiento y en la exactitud de las aplicaciones [9]. Además, resulta útil puesto que es capaz de detectar errores de memoria, sobrecargas, puede identificar factores que incrementen la velocidad de nuestro proyecto, minimiza el uso de memoria y permite además poder localizar y solventar posibles errores en el código del producto.

Su funcionamiento es simple. Valgrind instrumenta el código del proyecto mientras se ejecuta, seguidamente imprime un “commentary” el cual identifica de forma clara y concisa cada proceso implicado en su ejecución.

Además según la herramienta en uso y sus parámetros hacen que se varíe el nivel de instrumentación por lo que al ejecutar el código instrumentado podemos encontrar que este puede llegar a tardar de 4 a 50 veces más que la ejecución normal.

El principal cometido es crear perfiles de rendimiento, por lo que la herramienta que nos facilita esto es callgrind. Se encarga de crear perfiles que registran el historial de llamadas de función como un gráfico de llamadas.

Se caracteriza por su sencillez ya que únicamente se han de seguir unos pocos pasos como los que se muestran a continuación en los siguientes puntos:

- Se ha de compilar el programa para obtener un gráfico de llamadas significativo, para ello se han de tener los símbolos de depuración habilitados.
- Se ha de ejecutar el programa con valgrind –tool = callgrind ./programa con el fin de generar los archivos de perfil.
- Finalmente solo se han de analizar los datos de perfil por ejemplo con Kcachegrind.

OProfile

Es otro perfilador el cual se caracteriza por ser muy sencillo de usar y es más fácil de configurar que los dos anteriores [10].

Se caracteriza por usar hardware dedicado de monitorización de rendimiento de procesos para recuperar información sobre el kernel y los ejecutables del sistema.

Un ejemplo se tiene en lo que se refiere a la memoria, a la cantidad de solicitudes de cache y al número de interrupciones de hardware recibidas.

Este perfilador también puede ser usado con el fin de determinar el uso del procesador y las aplicaciones y servicios más utilizados.

Modulo 2. Parseo de código C++

En este modulo el principal cometido consiste en llevar a cabo las operaciones necesarias para conocer las funciones definidas dentro de los ficheros cpp que han sido modificados con el fin de almacenar estas en una base de datos o fichero csv el cual nos permita conocer que subsets de regresiones se han de ejecutar.

- Vim

Para ello se inició intentando realizar este parseo del código con Vim puesto que usando vim con ctags se puede generar un archivo de etiquetas con el fin de guardar todas las definiciones dentro del código. Se han de tener actualizadas las etiquetas cada vez que se genere el código lo cual lo hace imposible en nuestro proyecto puesto que contiene muchos archivos y lleva un gran tiempo parsear todos estos ficheros por lo que vim tomó incluso horas y en ocasiones se bloqueó por lo que se descartó como opción incluso ejecutándolo como tarea asíncrona [11].

- Cscope, Pyparser y CppHeader

Se usó y probó tambien estas bibliotecas/herramientas con el fin de extraer las cabeceras de las funciones sin éxito debido a numerosos factores como el volumen de nuestro código o la estructura que tiene.

- Pyparser: el problema que se encontró fue que no reconocía las funciones declaradas dentro de los namespaces y tampoco reconocía las librerías incluidas en los ficheros [12].
 -
 - Cscope: El problema que se encontró usando esta herramienta fue básicamente su compatibilidad con el entorno en el cual no pudo llegar a instalarse/usarse por lo que se descartó [13].
 -
 - CppHeaderParser: al igual que en Pyparser, se pudieron obtener los nombres de las funciones declaradas en los ficheros cpp pero no los nombres de las funciones situadas en sus namespaces, para darle solución al problema se intentó usar los token incluyendo los namespaces como declaraciones pero el resultado no fue bueno ya que no se obtuvieron los nombres de las funciones [14].

- Doxygen

Con el generador de documentación de código Doxygen se generó la correspondiente documentación en los ficheros del proyecto con el fin de a partir de un script en python poder acceder a ella por medio de la extracción de la información de los archivos html y xml generado usando para ello BeautifulSoup.

La siguiente imagen muestra una parte del código de como se hizo en una de las pruebas:

```

#!/usr/bin/python

from bs4 import BeautifulSoup
import urllib2
import unicodedata
import json

DOXY_PATH = "http://lonlnx48/ngdcs-doc/CM/dox/html/globals_func"
ALPH_LIST = map(chr, range(97, 123))

def get_functions_from_file(myf_name_list):
    filename_dictionary = {}
    for file_name in myf_name_list:
        functions_in_file = []
        func_dictionary = {}
        for word in ALPH_LIST:
            all_list = []
            if word == 'a':
                full_path = DOXY_PATH + ".html#index_a"
            else:
                full_path = DOXY_PATH + "_" + word + ".html#index_" + word
            try:
                soup = BeautifulSoup(urllib2.urlopen(full_path, "html.parser"))
                for item in soup.findall('tt'):
                    func_dictionary = {}
                    all_list.append(item.text)
            for element in all_list:
                aux = []
                aux = element.split('\n')
                func_dictionary[aux[0]] = aux[1]
            for key, value in func_dictionary.items():
                str_value = unicodedata.normalize('NFKD', value).encode('ascii', 'ignore')
                str_key = unicodedata.normalize('NFKD', key).encode('ascii', 'ignore')
                if file_name in str_value and file_name not in functions_in_file:
                    functions_in_file.append(str_key)
            except:
                pass

    filename_dictionary[file_name] = functions_in_file
    return filename_dictionary

def generate_db2_file(file_list):
    file_functions_dic = get_functions_from_file(file_list)
    dictionary_to_json = json.dumps(file_functions_dic)
    print "DB2 - Functions per cpp file: ", file_functions_dic
    with open('db2_Cpp_FNames.json', 'w') as outfile:
        outfile.write(dictionary_to_json)

def main():
    file_list = ['bsed.c', 'CPREMarshallManager.cpp', 'VoucherPrintStatus.cpp']
    generate_db2_file(file_list)

if __name__ == "__main__":
    main()

```

Ilustración 28. Ejemplo de parsing con Doxygen

El problema por el que no se ha usado finalmente esta solución ha sido debido a que se debía de generar continuamente los archivos de doxygen debido a que si no están actualizados se corre el riesgo de no obtener las funciones correctas. Por lo que no es muy fiable obtener información de los ficheros html si no están actualizados.

- [Clang](#)

Clang es upfront end de compilador para los lenguajes de programación C, C++, Objective C y Objective C++. Se caracteriza porque usa LLVM [15] a modo de back-end y ha sido parte de su ciclo. Se diseñó con el fin de ofrecer un reemplazo de GCC, es de código abierto y tiene compañías involucradas en su desarrollo como Google o Apple [16].

Es útil en nuestro proyecto debido a que incluye un analizador estático de software y varias herramientas de análisis de código las cuales han permitido en nuestras pruebas la extracción de las declaraciones de funciones de nuestros ficheros de código C++.

En las pruebas que se han realizado, ha permitido la obtención de estas declaraciones de funciones de forma rápida y concisa tras su instalación y configuración en el entorno por lo que finalmente se ha elegido.

4.4. Tecnología usada

- Perfiladores

Se procede a descartar Gprof debido a que se ha de reconstruir nuestra aplicación mientras que con valgrind (con callgrind) no se necesita reconstruirla con ningún indicador de compilación especial, por lo que se puede ejecutar cualquier tipo de ejecutable.

Otro punto positivo a favor de Valgrind supone que el programa ejecutado contiene la información de depuración la cual hace que se pueda obtener el gráfico de llamadas.

Por otro lado, la única desventaja de usar Valgrind sería que se produce una desaceleración en el perfilado por lo que puede ocurrir que sea de poca utilidad para aplicaciones de mayor o menor duración.

También se descarta Gprof debido a que la sobrecarga que produce llega a ser muy alta: estimada en 30-260%.

Y ademas siendo uno de los principales y mayores motivos por los que se termina de descartar Gprof es debido a que gprof no admite el perfilado de aplicaciones de subprocessos múltiples y tampoco puede crear perfiles de bibliotecas compartidas por lo que es completamente inadecuado para nuestro proyecto ya que tiene la gran mayoría de bibliotecas compartidas.

Si comparamos Oprofile con Gprof por que Gprof utiliza una llamada al sistema especial que periódicamente muestrea el proceso que se está ejecutando actualmente lo que significa que gprof solo sabe lo que sucede en tiempo de usuario para su proceso.

No será capaz de “ver” los cuellos de botella que se producen en el kernel o en las bibliotecas compartidas externas.

Esto puede causar que los resultados obtenidos en gprof se encuentren bastante sesgados para algunos tipos de cuellos de botella.

Gprof se encuentra limitado.

Sin embargo, por otra parte tenemos que Oprofile el cual lo “ve” todo ya que genera perfiles de todo el sistema que se activa en los eventos de hardware y va a llevar a cabo el registro de no solo los símbolos actuales que se ejecutan, sino que también va a registrar el proceso al que pertenecen o están involucrados. El único problema que se ha encontrado con Oprofile es que requiere una serie de permisos especiales (sudo) el cual no ha resultado ser un problema en nuestro proyecto.

Finalmente tras haber estudiado y puesto en pruebas los perfiladores, se ha decantado por usar Valgrind con callgraph o bien Oprofile por las ventajas que nos ofrecen con respecto a gprof.

- Neo4J

Para la visualización de las relaciones se ha usado Neo4J [17].

Es una de las muchas bases de datos orientadas a grafos (BDOG) las cuales ayudan a encontrar relaciones y dar sentido al entorno, como en nuestro caso dan sentido a las relaciones que hay entre funciones, librerías, y demás contenidos existentes en los sets de regresiones.

Se caracteriza por usar grafos con el fin de representar datos y las relaciones entre ellos. Un grafo en si es una representación gráfica la cual tiene vértices y aristas como principal elementos. Siguiendo esta definición se pueden encontrar distintos tipos de grafos como son los enumerados en los siguientes puntos:

- Grafos no dirigidos. En ellos los elementos nodo y los elementos relaciones pueden ser intercambiables, es decir, su relación puede ser interpretada en cualquier sentido.
- Grafos dirigidos. Al contrario que en los grafos anteriores, estos no son bidireccionales por defecto.
- Grafos con peso. En estos grafos las relaciones entre los nodos tienen una valoración numérica la cual permite realizar operaciones.
- Grafos con etiquetas. Estos grafos tienen incorporadas etiquetas la cuales definen los distintos vértices a la vez que también definen las relaciones que hay entre ellos.
- Grafos de propiedad. Es el tipo de grafo más complejo y se caracteriza porque son grafos con peso, con etiquetas y ademas se les pueden asignar propiedades a sus nodos o a sus relaciones.

Además, Neo4J se caracteriza por:

- Rendimiento. Tiene un rendimiento mejor que cualquier otra base de datos relacional o no relacional. Esto se debe principalmente a que aunque las consultas de datos aumente exponencialmente, su rendimiento no desciende.
- Agilidad. Tiene una gran agilidad en la gestión de datos. Si se intentase llevar al límite sus capacidades, se tendrían que superar todo el volumen de unos 34.000 millones de nodos , 34.000 millones de relaciones entre esos datos, 68.000 millones de propiedades y 32.000 tipos de relaciones.
- Flexibilidad y escalabilidad. Es una propiedad que las hace muy útil ya que para nuestra empresa por ejemplo se buscan estas características ya que se trabajan con grandes cantidades de datos. Son muy útiles porque esta base de datos orientada a grafos da la posibilidad de añadir mas nodos y relaciones a un grafo ya existente.

- Clang

Como se ha mencionado anteriormente, esta ha sido la herramienta elegida para realizar las labores de parseo del código C++, la cual ha permitido en todos los casos obtener resultados satisfactorios para todas las pruebas que se han realizado.

Es un compilador de código abierto para los lenguajes de la familia C. El origen de este compilador, que surge del proyecto LLVM. Clang está escrito en C++ y puede ser utilizado en sistemas operativos basados en Unix. Clang a día de hoy puede ser usado como una alternativa directa al compilador GCC, ofreciéndonos ciertas ventajas, entre las que destaca su diseño modular para el análisis estático del código.

Una de las mayores ventajas de Clang es el diseño modular de su desarrollo, que parece ser uno de los pilares desde la concepción del proyecto. Esto nos permite disponer de una API muy bien diseñada y poder emplearla para la construcción de nuestras propias herramientas para el análisis del código, de la misma manera en la que el frontend lo hace al compilarlo.

Es ventajoso su uso debido a que reduce el tiempo de compilación y el uso de memoria por lo que beneficia al análisis del código. Además, avisa de errores y warnings indicando de forma exacta el punto en el que se producen y la información asociada al mismo.

Otra gran ventaja de su uso se encuentra en que busca que las herramientas desarrolladas puedan integrarse de forma sencilla con IDEs (Entornos de desarrollo integrados). También por medio del uso de una licencia BSD permite que pueda ser empleada en productos comerciales y además trata de ajustarse de forma completa a los lenguajes de C y sus variantes.

Su instalación resulta muy sencilla ya que cuenta con un manual a modo de tutorial donde los pasos a seguir son muy simples e intuitivos [18].

5. Proceso de desarrollo y resultados

En este apartado se procede a realizar una descripción más detallada del software desarrollado. Para ello se va a llevar a cabo su explicación separando la aplicación web de la herramienta de tests.

5.1. Aplicación web para la creación de nuevos usuarios de regresión

La descripción de este producto se va a dividir en varios apartados los cuales son resumidos por los siguientes puntos:

- Introducción
- Funcionalidades
- Estructura del proyecto web
- Como usarlo

5.1.1. Introducción

La dashboard de usuarios de regresiones es parte del paquete de la compañía llamado INT_Dashboard el cual se encuentra en (https://rndwww.nce.amadeus.net/git/projects/DCSI/repos/int_dashboard/browse).

Esta dashboard en concreto permite crear de forma sencilla e intuitiva un nuevo usuario con los permisos necesarios para ejecutar las pruebas funcionales. Para ello únicamente ha de añadir de forma obligatoria su identificador el cual se corresponde con su Unix Id y la máquina desde la que trabaja. Unicamente introduciendo estos datos se creará el usuario con un simple click.

También existe el panel de opciones avanzadas en el cual se pueden insertar más datos del usuario como el nombres que se le va a dar en la base de datos, la base de datos en la que se va a crear o el numero de puerto.

Esta sería la primera vista que tiene el usuario de nuestra aplicación, en la cual además puede redirigirse a la vista de “listar usuarios” donde verá la lista entera de todos los usuarios con su correspondiente información.

Además desde esta vista inicial puede también llevar a cabo las sincronizaciones del fichero de usuarios con las bases de datos, también puede exportar los usuarios a un fichero csv o importar los usuarios a la base de datos desde el fichero users.conf alojado en Bitbucket a modo de segunda fuente.

En la segunda vista, el usuario puede listar todos los usuarios creados filtrándolos por nombre, base de datos, puerto o por alias en la base de datos por ejemplo. Para ello, únicamente ha de usar o bien los campos de filtros o el filtro general.

Además el usuario también puede llevar a cabo una o varias migraciones a distintas bases de datos, puede a su vez modificar el contenido de los usuarios y también puede eliminarlos.

A continuación en la siguiente imagen se muestra la ventana de creación de usuario.

The screenshot shows a web browser window with the following details:

- Address Bar:** Shows the URL `ionlnx48:40000/new_comer/create_user/#`.
- Header:** Displays "New Comers Dashboard | Create" and "Log in - Help -".
- Form Title:** "Create regressions users".
- Input Fields (Unix ID):** A text input field containing "stiago".
- Input Fields (Machine):** A text input field containing "ionlnx48".
- Section Header:** "Advanced options -".
- Input Fields (DB Username):** A text input field containing "stiago".
- Input Fields (Database):** A dropdown menu showing "LONDB04".
- Input Fields (Port):** A text input field containing "175".
- Buttons:** A "Send" button at the bottom right of the form area.
- Footer Buttons:** "List users", "Import from users repository", "Synchronize changes", and "Export to CSV".
- Page Footer:** "New comers dashboard © 2017".

Ilustración 29. Ventana de creación de usuarios.

5.1.2. Funcionalidades

- **Crear-usuario.**

El principal cometido de esta funcionalidad consiste en básicamente crear un nuevo usuario de regresión el cual pueda ejecutar estas pruebas funcionales. Para ello simplemente se ha de añadir el Unix Id del usuario en particular y la máquina en la que trabaja. Esto genera automáticamente el nombre en las bases de datos de los peak 1 y peak 2, la base de datos y también asigna de forma automática e inteligente un puerto de forma que no cree sobrecarga de los puertos. Por defecto, asigna el puerto con menor número de ocurrencias evitando con ello las múltiples colisiones, es decir, asigna el puerto menos usado de la lista disponible de puertos disponibles.

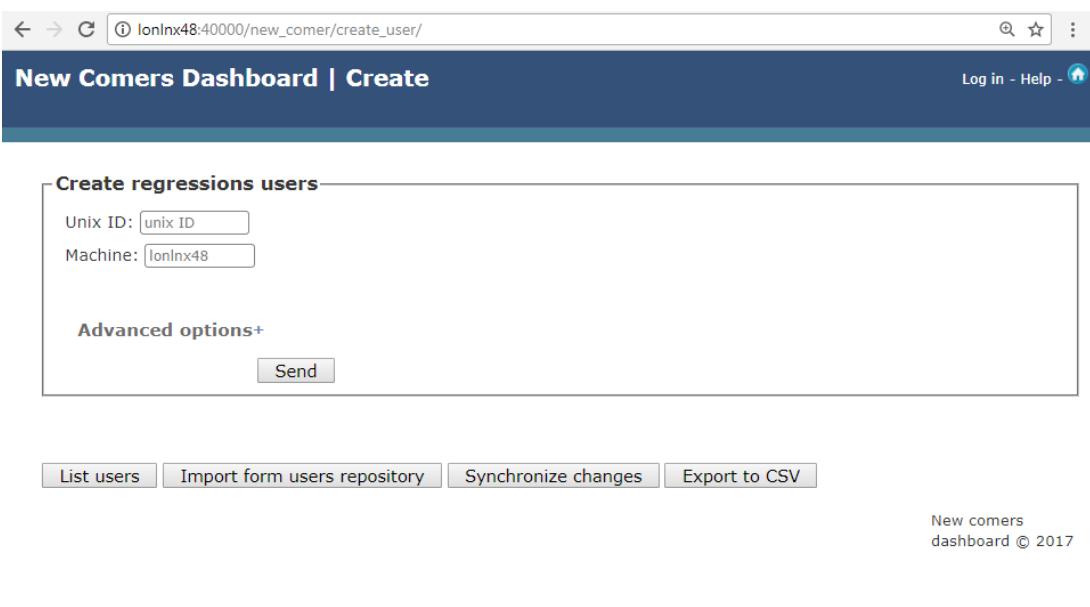


Ilustración 30. Crear usuario.

- **Listado**

Esta vista permite al usuario de SFA – INT (Software Framework Architecture - Integration) listar la información de todos los usuarios de regresión creados.

Específicamente muestra la información referente a sus Unix Id, los nombres que tienen en la base de datos del peak 1 y peak 2, la base de datos en la que se ha creado y el numero de puerto que se les ha asignado. También en esta vista se encuentran las herramientas necesarias para llevar a cabo la manipulación de los usuarios como bien puede ser la tarea de modificación, borrado, migrado de base de datos, etc.

The screenshot shows a web-based application interface titled 'New Comers Dashboard | Comers's List'. The main area displays a table of user data with columns: ID, Unix ID, DB Username Peak 1, DB Username Peak 2, Database, Port, Update, Delete, and a grid icon. The table contains approximately 30 rows of user information. To the right of the table is a sidebar titled 'Options' with three items: 'Select / Deselect All', 'Migrate to: LONDB04', and 'General filter:'. At the bottom of the page are buttons for 'List users', 'Import from users repository', 'Synchronize changes', and 'Export to CSV'.

ID	Unix ID	DB Username Peak 1	DB Username Peak 2	Database	Port	Update	Delete	Grid
4013	vsantiago	vsantiago	vsantiago_p2	LONDB04	174			
3933	rdagenais	rdagenais	rdagenais_p2	LONDB04	203			
3932	rblusr2	rblusr2	rblusr2_p2	LONDB04	177			
3931	rblusr1	rblusr1	rblusr1_p2	LONDB04	177			
3930	cdevautour	cdevautour	cdevautour_p2	LONDB04	176			
3929	jacopo.biscella	jacopo.biscella	jacopo.biscella_p2	NCMUSR04LS	153			
3928	daniel.hurtado	daniel.hurtado	daniel.hurtado_p2	LONDB04	240			
3927	pannamalai	pannamalai	pannamalai_p2	NCMUSR02LS	227			
3926	anand.palani	anand.palani	anand.palani_p2	LONDB04	213			
3925	psathineni	psathineni	psathineni_p2	LONDB04	212			
3924	ame louki	ame louki	ame louki_p2	LONDB04	211			
3923	suganya.chandrasek	suganya.chandrasek	suganya.chandrasek_p2	LONDB04	209			
3922	guido.maresca	guido.maresca	guido.maresca_p2	NCMUSR04LS	204			
3921	hamida.yadav	hamida.yadav	hamida.yadav_p2	LONDB04	195			
3934	oleksii.novikov	oleksii.novikov	oleksii.novikov_p2	LONDB04	233			
3950	ssandhu	ssandhu	ssandhu_p2	LONDB04	111			
3936	rupali.bendale	rupali.bendale	rupali.bendale_p2	NCMUSR05LS	23			
3949	ssingh	ssingh	ssingh_p2	LONDB04	190			
3948	janos.olah	janos.olah	janos.olah_p2	LONDB04	38			

Ilustración 31. Vista de listar usuarios.

- **Modificar / Actualizar usuario.**

Esta acción consiste básicamente en actualizar el contenido de un usuario como bien puede ser el puerto, la base de datos o el alias entre otros.

New Comers Dashboard | Update

Log in - Help - [Home](#)

Update comer

ID: 4013
Unix ID:
DB Username Peak 1:
DB Username Peak 2:
Database: LONDB04 (current selection)
Port:

Ilustración 32. Actualizar información de usuario.

- **Eliminar usuario.**

Consiste en borrar un usuario seleccionado de la base de datos, para ello únicamente se ha de marcar eliminar sobre la ficha del usuario en cuestión. Esta acción nos llevará a una ventana de confirmación en la que finalmente se elegirá si se quiere eliminar o se descarta la operación.

New Comers Dashboard | Update

Log in - Help - [Home](#)

Delete comer

ID: 4013
Unix ID: **vsantiago**
DB Username Peak 1: **vsantiago**
DB Username Peak 2: **vsantiago_p2**
Database: **LONDB04**
Port: **174**

Ilustración 33. Eliminar usuario.

- **Migrar usuario.**

Esta funcionalidad permite migrar uno o varios usuarios de una base de datos a otra simplemente seleccionando las casillas de los usuarios que se quieran migrar y seguidamente se ha de seleccionar la base de datos a la que se desea hacer la migración para finalmente hacer click en el botón “Migrate”.

Options:

Select / Deselect All

▪ Migrate to: LONDB04 ▾

▪ General filter

- LONDB04
- LOCALDB
- NCMUSR02LS
- NCMUSR03LS**
- NCMUSR04LS
- NCMUSR05LS

Ilustración 34. Opción de migrar usuarios

- **Filtros.**

Esta opción se encuentra contenida en la vista de “Lista de usuarios”. Se encuentran disponibles dos tipos de filtros: filtro general y filtro por campos.

- Filtro General: esta herramienta da la posibilidad de realizar un filtrado de toda la información existente en la lista de usuarios. Simplemente se ha de insertar un contenido en la caja de búsqueda y esta sera filtrada por cada uno de los campos.

Not secure | lonlnx48:40000/new_comer/list_user/

Regression users dashboard | Comers's List

Log in - Help -

Latest regression users

ID	Unix ID	DB Username Peak 1	DB Username Peak 2	Database	Port	Update	Delete
3933	rdagenais	rdagenais	rdagenais_p2	LONDB04	203		
3934	oleksii.novikov	oleksiinovikov	oleksiinovikov_p2	LONDB04	133		
3895	hsenussi	hsenussi	hsenussi_p2	LONDB04	33		
3933	jjoose	jjoose	jjoose_p2	LONDB04	63		
3933	dimitrios.vikeloudas	dimitriosvikeloudas	dimitriosvikeloudas_p2	LONDB04	43		
3838	pawel.rozenberg	pawelrozenberg	pawelrozenberg_p2	NCMUSR02LS	133		

Options:

- Select / Deselect All
- Migrate to: LONDB04
- General filter:

Ilustración 35. Ejemplo de filtro general.

- Filtro por campos: este filtro permite buscar un contenido en un campo específico.

Regression users dashboard | Comers's List

Latest regression users

ID	Unix ID	DB Username Peak 1	DB Username Peak 2	Database	Port	Update	Delete
	vs						
4013	vsantiago	vsantiago	vsantiago_p2	LONDB04	174		
3997	vsoomaneyvijaykumar	vsoomaneyvijaykumar	vsoomaneyvijaykumar_p2	LONDB04	68		

Ilustración 36. Ejemplo de filtro de usuarios por Unix ID

- **Import desde el repositorio de usuarios.**

Esta funcionalidad permite por medio del click en su botón, la importación o eliminado de los usuarios que se hayan creado o eliminado en el fichero users.conf en el repositorio de Bitbucket. La importación se realizará directamente en nuestra base de datos MySQL.

```
Unix_id sharmap already set.  
Unix_id ssatwah already set.  
Unix_id acurras already set.  
New user: qbernard  
New user: mgocieva  
New user: mvaitkus  
New user: skopcsa  
New user: lhopkins  
[09/Feb/2018 14:54:47] "GET /new_comer/parse_file/ HTTP/1.1" 302 0  
[09/Feb/2018 14:54:48] "GET /new_comer/list_user/ HTTP/1.1" 200 283917
```

Ilustración 37. Resultado de la importación de nuevos usuarios.

- **Sincronización de cambios.**

Se encarga de generar el fichero users.conf con las correspondientes modificaciones ya sean de inserción, actualizado o eliminado de usuarios. Para llevar a cabo la correspondiente inserción, actualizado o borrado se han de hacer manteniendo los cambios en Mercurial realizando esta sincronización con el fin de tener el mismo contenido en los distintos sitios.

- **Exportar a un fichero CSV.**

Nos permite la dashboard por medio de esta funcionalidad poder exportar el listado de todos los usuarios con sus características o datos hacia un fichero CSV el cual puede ser visualizado por ejemplo en excel.

A continuación en la siguiente imagen se muestra un ejemplo del tipo de fichero csv generado.

A	B	C	D	E	F	
1	ID	Unix id	DB Username Peak 1	DB Username Peak 2	Database	Port
2	4013	vsantiago	vsantiago	vsantiago_p2	LONDB04	174
3	3933	rdagenais	rdagenais	rdagenais_p2	LONDB04	203
4	3932	rblusr2	rblusr2	rblusr2_p2	LONDB04	177
5	3931	rblusr1	rblusr1	rblusr1_p2	LONDB04	177
6	3930	cdevautour	cdevautour	cdevautour_p2	LONDB04	176
7	3929	jacopo.biscella	jacopobiscella	jacopobiscella_p2	NCMUSR04LS	153
8	3928	daniel.hurtado	danielhurtado	danielhurtado_p2	LONDB04	240
9	3927	pannamalai	pannamalai	pannamalai_p2	NCMUSR02LS	227
10	3926	anand.palani	anandpalani	anandpalani_p2	LONDB04	213
11	3925	psathineni	psathineni	psathineni_p2	LONDB04	212
12	3924	amelouki	amelouki	amelouki_p2	LONDB04	211
13	3923	suganyaachandraseka	suganyaachandraseka	suganyaachandraseka_p2	LONDB04	209
14	3922	guido.maresca	guidomaresca	guidomaresca_p2	NCMUSR04LS	204
15	3921	hamida.yadav	hamidayadav	hamidayadav_p2	LONDB04	195

Ilustración 38. Ejemplo de fichero csv generado.

5.1.3. Estructura del proyecto.

Para su creación inicialmente necesitamos añadir las siguientes líneas en el siguientes ficheros de configuración:

- Settings.py file: En este fichero se ha de añadir el nombre de nuestra aplicación, en nuestro caso 'new_comer', en el apartado "/INSTALLED_APPS".
- Urls.py: Se ha de añadir aquí nuestra url en "urlpatterns" de forma similar a `url(r'^new_comer/', include(new_comer.urls))` y además se ha de introducir la importación "import new_comer.urls".

Seguidamente se ha creado el directorio "new_comer" en el cual podemos encontrar los ficheros de nuestra dashboard los cuales son:

- `__init__.py`
- `Admin.py`
- `Apps.py`
- `Forms.py`
- `Models.py`
- `Tools.py`
- `Urls.py`
- `Views.py`

Todos los métodos funcionales de inserción, actualización de contenidos, borrado y visualización (CRUD) se hayan implementados en el fichero `views.py` en el cual se puede encontrar como se ha comentado anteriormente en el directorio `/int_dashboard/new_comer`.

En el fichero `models.py` se encuentra definida toda la información sobre nuestro modelo de datos donde están definidos los campos necesarios con sus tipos para almacenar la información.

Los demás métodos auxiliares y herramientas como el método de asignación inteligente de puertos se encuentran en el fichero `tools.py` el cual a modo de otro módulo los almacena.

Las urls necesarias para conectar la funcionalidad con cada vista se encuentran en el fichero `urls`.

5.2. Herramienta para pruebas de regresión

Esta herramienta como se ha descrito anteriormente, va a estar dividida en módulos para su mejor desarrollo.

Módulo 0 – Generar ficheros csv a partir de los escenarios (tests de regresión) - División de regresiones en subsets

Este módulo se corresponde con la creación de los subsets de regresión de forma local.

El funcionamiento y procedimiento de este módulo se puede describir con los pasos que se muestran en la siguiente imagen.

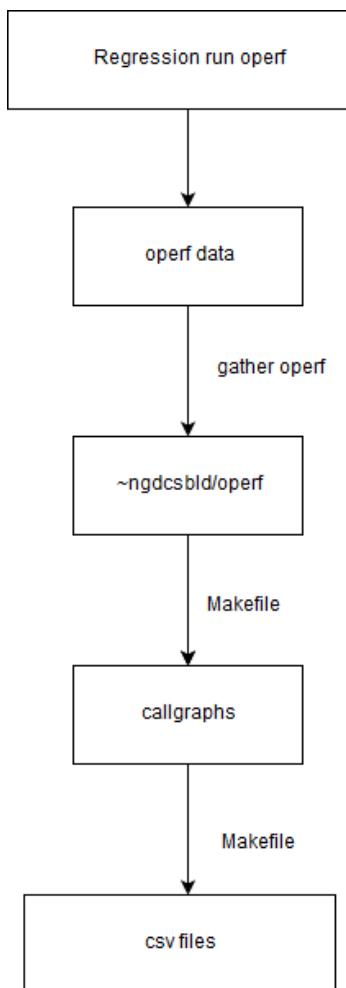


Ilustración 39. Funcionamiento del módulo 0 donde se generan los ficheros csv.

Lanzando regresiones con operf en local

En esta parte del proyecto se documentan los pasos seguidos para lanzar la suite de regresiones bajo el perfil de Operf a la vez que podremos también recolectar los resultados con el fichero de gather.

Se ha de mencionar que operf profiler es parte del paquete oprofile el cual podemos encontrar en el siguiente enlace: <http://oprofile.sourceforge.net/news/>.

Las máquinas de desarrollo de nuestro Sistema de control de salidas (DCS) han sido dotadas con el paquete oprofile en su versión v1.1.0 la cual se ha instalado como parte de las herramientas localizadas en nuestro paquete ngd_softs.

Los test de regresión pueden ser lanzados con operf inicialmente estableciendo la variable PROFILER como sigue a continuación:

```
ntt PROFILER=oprofile config; # Configure with oprofile  
ntt boot; # Boot the backends
```

El siguiente paso es inspeccionar los procesos y ver que el proceso otf_be tiene un fork desde los procesos operf.

```
> ps fU $USER  
  
19698 ? Ssl 0:04 sei_agent -nm 01_CML -tc agent.traceconf  
19911 ? S1 0:13 \_ otf_fe v14.0.0.36 : 01_CML FE_CML  
19915 ? S1 0:08 \_ otf_cs v14.0.0.36 : 01_CML CS_FE_CML  
19919 ? S 0:00 \_ operf -g -d ../log/oprofile.out.RGR_LST.1498062589 /data/mwdeldev/ngdpack-14/mdw/Pack/14  
20860 ? S1 4:21 | \_ otf_be v14.0.0.36 : 01_CML RGR_LST_CML #1 [processing service: APALRQ, service converge  
20861 ? S 0:00 | \_ operf -g -d ../log/oprofile.out.RGR_LST.1498062589 /data/mwdeldev/ngdpack-14/mdw/Pac  
20862 ? S 0:25 | \_ operf -g -d ../log/oprofile.out.RGR_LST.1498062589 /data/mwdeldev/ngdpack-14/mdw/Pac  
19922 ? S 0:00 \_ operf -g -d ../log/oprofile.out.RGR_SYN.1498062589 /data/mwdeldev/ngdpack-14/mdw/Pack/14  
20855 ? S1 0:16 | \_ otf_be v14.0.0.36 : 01_CML RGR_SYN_CML #1 [idle]  
20856 ? S 0:00 | \_ operf -g -d ../log/oprofile.out.RGR_SYN.1498062589 /data/mwdeldev/ngdpack-14/mdw/Pac  
20858 ? S 0:00 | \_ operf -g -d ../log/oprofile.out.RGR_SYN.1498062589 /data/mwdeldev/ngdpack-14/mdw/Pac  
19924 ? S1 1:21 \_ otf_fe v14.0.0.36 : 01_CML LQS_CML  
18967 ? Ssl 0:12 sei_master -tc mag.traceconf
```

Ilustración 40. Procesos Operf

Los datos de operf son almacenados en el directorio log, específicamente dentro de su subdirectorio oprofile.out.<be_name>.<timestamp>, por ejemplo:

log/oprofile.out.RGR_SYN_1498062589

Cuando se finalice el testeo, se ha de asegurar de cerrar la conexión por así decirlo del nodo para finalizar el profiling. Esto se realiza normalmente mediante el comando:

ntt vsy

Posibles errores:

Se ha de revisar en caso de error los ficheros log/agent.stdout y log/agent.stderr en busca de errores si el backend no se inicia.

Call-graph:

Los datos generados por operf pueden ser interpretados mediante oreport. En el siguiente enlace se pueden visualizar algunos ejemplos de cómo oreport los produce.

Para general el call graph se ha usado el siguiente comando:

```
opreport --session-dir log/oprofile.out.RGR_SYN.1498062589  
--callgraph --long-filenames > RGR_SYN_call_graph
```

Si eliminamos la opción --log-filenames, el nombre de la librería es truncada hacia su propio nombre de fichero en el jobrep cache el cual es solo una cadena de dígitos.

El call-graph finalmente es una serie de tablas las cuales listan los siguientes apartados:

- Call count: hace referencia al número de veces que una función es llamada.
- Call percentage: cuantas funciones son.
- Image: cuan binaria es.
- Function names: nombre de la función.

En cada bloque tenemos que una de las filas no se encuentra indentada. Esta viene a ser la función de interés de este bloque.

A continuación tenemos un ejemplo de la tabla que se tiene para por ejemplo:

```
SYN::PaxManager::synPax(SYN::IreqType)
```

3	0.0013	/data/mwdeldev/ngdpack-14/ngd/core/ngobe/14-4-4-3/lib/x86_64-2.6.16-v1/Debug/libcomKitOBE.so.14.4.4
234388	99.9987	/remote/vtmp2/ngdcs.jobrep/cmapp/ngsyn/lib/x86_64-2.6.16-v1/Debug/ora/libcomKitSYN.so.0.0.0/149803
5	1.6e-04	/remote/vtmp2/ngdcs.jobrep/cmapp/ngsyn/lib/x86_64-2.6.16-v1/Debug/ora/libcomKitSYN.so.0.0.0/14980338
234289	99.9561	/remote/vtmp2/ngdcs.jobrep/cmapp/ngsyn/lib/x86_64-2.6.16-v1/Debug/ora/libcomKitSYN.so.0.0.0/149803
61	0.0260	/remote/vtmp2/ngdcs.jobrep/cmapp/ngsyn/lib/x86_64-2.6.16-v1/Debug/ora/libcomKitSYN.so.0.0.0/149803
12	0.0051	/remote/vtmp2/ngdcs.jobrep/cmapp/ngsyn/lib/x86_64-2.6.16-v1/Debug/ora/libcomKitSYN.so.0.0.0/149803
6	0.0026	/remote/vtmp2/ngdcs.jobrep/cmapp/ngsyn/lib/x86_64-2.6.16-v1/Debug/ora/libcomKitSYN.so.0.0.0/149803
5	0.0021	/remote/vtmp2/ngdcs.jobrep/cmapp/ngsyn/lib/x86_64-2.6.16-v1/Debug/ora/libcomKitSYN.so.0.0.0/149803
3	0.0013	/data/mwdeldev/ngdpack-14/ngd/core/ngdbo/14-4-4-3/lib/x86_64-2.6.16-v1/Debug/libcomKitDBO.so.14.4.4
3	0.0013	/data/mwdeldev/ngdpack-14/ngd/core/ngdbo/14-4-4-3/lib/x86_64-2.6.16-v1/Debug/libcomKitDBO.so.14.4.4
3	0.0013	/remote/vtmp2/ngdcs.jobrep/cmapp/ngsyn/lib/x86_64-2.6.16-v1/Debug/ora/libcomKitSYN.so.0.0.0/149803
2	8.5e-04	/data/mwdeldev/ngdpack-14/ngd/core/ngdbo/14-4-4-3/lib/x86_64-2.6.16-v1/Debug/libcomKitDBO.so.14.4.4
2	8.5e-04	/remote/vtmp2/ngdcs.jobrep/cmapp/ngsyn/lib/x86_64-2.6.16-v1/Debug/ora/libcomKitSYN.so.0.0.0/149803
2	8.5e-04	/remote/vtmp2/ngdcs.jobrep/cmapp/ngsyn/lib/x86_64-2.6.16-v1/Debug/ora/libcomKitSYN.so.0.0.0/149803
1	4.3e-04	/proc/kallsyms ret_from_intr
1	4.3e-04	/remote/vtmp2/ngdcs.jobrep/cmapp/ngsyn/lib/x86_64-2.6.16-v1/Debug/ora/libcomKitSYN.so.0.0.0/149803
1	4.3e-04	/remote/vtmp2/ngdcs.jobrep/cmapp/ngsyn/lib/x86_64-2.6.16-v1/Debug/ora/libcomKitSYN.so.0.0.0/149803
1	4.3e-04	/remote/vtmp2/ngdcs.jobrep/cmapp/ngsyn/lib/x86_64-2.6.16-v1/Debug/ora/libcomKitSYN.so.0.0.0/149803

Ilustración 41. Tabla para SYN

A continuación, tras plasmar e introducir brevemente el funcionamiento de operf, se va a pasar a lanzar la entera suit de regresiones con operf.

Se puede realizar un trigger de las regresiones usando la configuración de operf y la cola. El comando viene a ser el siguiente:

```
/projects/ngdcsbld/cronbin/ngpost.sh operf tst:<branch>:<reléase>:operf:: validate_all RELEASE=<reléase> TESTTYPE=operf BRANCH=<branch>
```

En ello indicamos cada rama y cada release de nuestros proyectos.

El proceso se puede ir visualizando en el directorio de colas de operf como muestra la siguiente captura de pantalla. Únicamente se debe de ir a dicho directorio y listar su contenido.

Directorio: ngdcsbld/queues/operf

```

lonlnx48.lon.amadeus.net - PuTTY
.(~ngdcsbld/queues/operf)
--> tree
.
-- HISTORY
| -- DETAIL.2018-04-25-09:26:02-UTC.tst:PRD_2018Q3:59-0-41:operf:::ncmde103_db1.ok
| -- DETAIL.2018-04-25-09:26:02-UTC.tst:PRD_2018Q3:59-0-41:operf:cmtec-ngxdcrgr:e2e.ncmde101_db5.ok
| -- DETAIL.2018-04-25-09:26:03-UTC.tst:PRD_2018Q3:59-0-41:operf:cmtec-ngxdcrgr:generated.ncmde101_db8.failed
| -- DETAIL.2018-04-25-09:27:02-UTC.tst:PRD_2018Q3:59-0-41:operf:cmtec-ngrstrgr:liverestore_cpr.ncmde102_db4.failed
| -- DETAIL.2018-04-25-09:27:02-UTC.tst:PRD_2018Q3:59-0-41:operf:cmtec-ngrstrgr:liverestore_cpr_misc.ncmde111_db4.ok
| -- DETAIL.2018-04-25-09:27:03-UTC.tst:PRD_2018Q3:59-0-41:operf:cmtec-ngrstrgr:fli.ncmde102_db8.ok
| -- DETAIL.2018-04-25-09:27:03-UTC.tst:PRD_2018Q3:59-0-41:operf:cmtec-ngrstrgr:generated.ncmde111_db2.ok
| -- DETAIL.2018-04-25-09:27:04-UTC.tst:PRD_2018Q3:59-0-41:operf:cmtec-ngrstrgr:liverestore_ncmde103_db2.ok
| -- DETAIL.2018-04-25-09:27:04-UTC.tst:PRD_2018Q3:59-0-41:operf:cmtec-ngrstrgr:cpr_reg_cfr.ncmde101_db2.ok
| -- DETAIL.2018-04-25-09:28:02-UTC.tst:PRD_2018Q3:59-0-41:operf:cmtec-ngrlvrgr:cpr_pushpax.ncmde102_db5.ok
| -- DETAIL.2018-04-25-09:31:02-UTC.tst:PRD_2018Q3:59-0-41:operf:cmtec-ngrlvrgr:cpr_purge.ncmde103_db1.ok
| -- DETAIL.2018-04-25-09:37:02-UTC.tst:PRD_2018Q3:59-0-41:operf:cmapp-ngidcrgr:mig.ncmde111_db4.ok
| -- DETAIL.2018-04-25-09:38:03-UTC.tst:PRD_2018Q3:59-0-41:operf:cmapp-ngidcrgr:merge.ncmde102_db4.ok
| -- DETAIL.2018-04-25-09:39:02-UTC.tst:PRD_2018Q3:59-0-41:operf:cmapp-ngidcrgr:edi7.ncmde102_db5.ok
| -- DETAIL.2018-04-25-09:39:03-UTC.tst:PRD_2018Q3:59-0-41:operf:cmapp-ngidcrgr:edi6.ncmde103_db1.ok
| -- DETAIL.2018-04-25-09:40:04-UTC.tst:PRD_2018Q3:59-0-41:operf:cmapp-ngidcrgr:edi5.ncmde101_db5.ok
| -- DETAIL.2018-04-25-09:42:03-UTC.tst:PRD_2018Q3:59-0-41:operf:cmapp-ngidcrgr:edi4.ncmde103_db2.ok
| -- DETAIL.2018-04-25-09:43:02-UTC.tst:PRD_2018Q3:59-0-41:operf:cmapp-ngidcrgr:edi3.ncmde102_db8.ok
| -- DETAIL.2018-04-25-09:45:03-UTC.tst:PRD_2018Q3:59-0-41:operf:cmapp-ngidcrgr:edi1.ncmde111_db4.ok
| -- DETAIL.2018-04-25-09:45:04-UTC.tst:PRD_2018Q3:59-0-41:operf:cmapp-ngidcrgr:edi2.ncmde101_db8.ok
| -- DETAIL.2018-04-25-10:05:03-UTC.tst:PRD_2018Q3:59-0-41:operf:cmapp-ngidcrgr:cpt.ncmde101_db2.ok
| -- DETAIL.2018-04-25-10:11:03-UTC.tst:PRD_2018Q3:59-0-41:operf:cmapp-ngidcrgr:cry.ncmde102_db4.ok
| -- DETAIL.2018-04-25-14:48:02-UTC.regression_followup:gather_operf_data:PRD_2018Q3:59-0-41:operf.ncmde102_db4.ok
| -- DETAIL.2018-04-25-14:48:03-UTC.gather_operf_data:_vtmp_ngdcsbld_PRD_2018Q3_59-0-41_internal_cmtec_ngrstrgr_liverestore_cpr.ncmde102_db8.ok
| -- DETAIL.2018-04-25-14:48:03-UTC.gather_operf_data:_vtmp_ngdcsbld_PRD_2018Q3_59-0-41_internal_cmtec_ngxdcrgr_e2e.ncmde101_db5.ok
| -- DETAIL.2018-04-25-14:49:02-UTC.gather_operf_data:_vtmp_ngdcsbld_PRD_2018Q3_59-0-41_internal_cmtec_ngrlvrgr_cpr_pushpax.ncmde101_db8.ok
| -- DETAIL.2018-04-25-14:49:02-UTC.gather_operf_data:_vtmp_ngdcsbld_PRD_2018Q3_59-0-41_internal_cmtec_ngrstrgr_liverestore.ncmde102_db4.ok
| -- DETAIL.2018-04-25-14:49:03-UTC.gather_operf_data:_vtmp_ngdcsbld_PRD_2018Q3_59-0-41_internal_cmapp_ngidcrgr_merge.ncmde101_db5.ok
| -- DETAIL.2018-04-25-14:49:03-UTC.gather_operf_data:_vtmp_ngdcsbld_PRD_2018Q3_59-0-41_internal_cmapp_ngidcrgr_mig.ncmde102_db5.ok
| -- DETAIL.2018-04-25-14:49:03-UTC.gather_operf_data:_vtmp_ngdcsbld_PRD_2018Q3_59-0-41_internal_cmtec_ngxdcrgr_generated.ncmde111_db4.ok
| -- DETAIL.2018-04-25-14:51:03-UTC.gather_operf_data:_vtmp_ngdcsbld_PRD_2018Q3_59-0-41_internal_cmapp_ngidcrgr_edi4.ncmde102_db4.ok
-- PIDS
| -- ncmde101_db2.lonlnx28.pid
| -- ncmde101_db5.lonlnx35.pid
| -- ncmde101_db8.lonlnx30.pid
| -- ncmde102_db4.lonlnx46.pid
| -- ncmde102_db5.lonlnx47.pid
| -- ncmde102_db8.lonlnx48.pid
| -- ncmde103_db1.lonlnx51.pid
| -- ncmde111_db2.lonlnx34.pid
| -- ncmde111_db4.lonlnx54.pid
-- QUEUE
| -- gather_operf_data:_vtmp_ngdcsbld_PRD_2018Q3_59-0-41_internal_cmapp_ngidcrgr_edi3.ngrequest
| -- gather_operf_data:_vtmp_ngdcsbld_PRD_2018Q3_59-0-41_internal_cmapp_ngidcrgr_edi5.ngrequest
| -- gather_operf_data:_vtmp_ngdcsbld_PRD_2018Q3_59-0-41_internal_cmtec_ngrstrgr_generated.ngrequest
| -- lonlnx29.blocked
| -- lonlnx53.blocked
-- SUMMARY.LOG
| -- ncmde101_db2.DETAIL.LOG
| -- ncmde101_db4.DETAIL.LOG
| -- ncmde101_db5.DETAIL.LOG
| -- ncmde101_db8.DETAIL.LOG
| -- ncmde102_db4.DETAIL.LOG
| -- ncmde102_db5.DETAIL.LOG
| -- ncmde102_db8.DETAIL.LOG
| -- ncmde103_db1.DETAIL.LOG
| -- ncmde103_db2.DETAIL.LOG
| -- ncmde111_db2.DETAIL.LOG
| -- ncmde111_db4.DETAIL.LOG

```

Ilustración 42. Regresiones por PIDS, en COLA y realizadas en HISTORY

Recogida de los datos generados por operf después de lanzar un test de regresión

Una vez que el lanzamiento de los test de regresión haya sido completado, los datos generados por operf son almacenados usando nuestra herramienta cronbin con el siguiente comando:

```
/projects/ngdcsbld/cronbin/ngpost.sh operf  
regression_followup:gather_operf_data:<branch>:<release>:operf  
regression_followup TASK=gather_operf_data RELEASE=<release>  
QUEUE=operf BRANCH=<branch> TO_QUEUE=operf
```

Este comando lo que hace en si es encolar nuestra tarea para analizar el historial de la regresión y encolar la tarea gather_operf_data para cada test de regresión que haya sido lanzado.

De lo que se esta encargando nuestra tarea llamada gather_operf_data es de almacenar todos los directorios de log/oprofile.out<BE_name>.<timestamp> en el directorio ~ngdcsbld/operf para su procesamiento.

Para ello se usa un Makefile con el fin de transformar cada uno de ellos en call-graphs. A continuación se tiene un ejemplo del scrip usado:

Makefile

```
SOURCES := $(wildcard ../../55-0-79_PRIA+55-0-79-1/oprofile.out*)  
CALL_GRAPHS := $(patsubst ../../55-0-79_PRIA+55-0-79-1/%,%_call_graph,${SOURCES})  
all : ${CALL_GRAPHS}  
  
%_call_graph : ../../55-0-79_PRIA+55-0-79-1/%  
    opreport --session-dir $^ --callgraph --long-filenames > $@
```

Ilustración 43. Ejemplo de Makefile usado

```

lonlnx48.lon.amadeus.net - PuTTY
oprofile.out.RGR_AXF.1498047579  oprofile.out.RGR_CPO.1498049072  oprofile.out.RGR_FLI.1
.-(~ngdcsbld/operf/PRD_2017Q3+55-0-83)-----
`-> ls -ltr
total 680
drwxr-xr-x 3 ngdcsbld ngdcsbld 80 Jun 21 2017 oprofile.out.RGR_TCH.1498047579
drwxr-xr-x 3 ngdcsbld ngdcsbld 80 Jun 21 2017 oprofile.out.RGR_CPR.1498048828
drwxr-xr-x 3 ngdcsbld ngdcsbld 1024 Nov 15 09:41 oprofile.out.RGR_ACC.1498047311
drwxr-xr-x 3 ngdcsbld ngdcsbld 1024 Nov 15 09:41 oprofile.out.RGR_ACC.1498047474
drwxr-xr-x 3 ngdcsbld ngdcsbld 1024 Nov 15 09:41 oprofile.out.RGR_ACC.1498047579
drwxr-xr-x 3 ngdcsbld ngdcsbld 1024 Nov 15 09:41 oprofile.out.RGR_ACC.1498048828
drwxr-xr-x 3 ngdcsbld ngdcsbld 1024 Nov 15 09:41 oprofile.out.RGR_ACR.1498047311
drwxr-xr-x 3 ngdcsbld ngdcsbld 1024 Nov 15 09:42 oprofile.out.RGR_ACR.1498047420
drwxr-xr-x 3 ngdcsbld ngdcsbld 1024 Nov 15 09:42 oprofile.out.RGR_ACR.1498047474
drwxr-xr-x 3 ngdcsbld ngdcsbld 1024 Nov 15 09:42 oprofile.out.RGR_ACR.1498047508
drwxr-xr-x 3 ngdcsbld ngdcsbld 1024 Nov 15 09:42 oprofile.out.RGR_ACR.1498047579
drwxr-xr-x 3 ngdcsbld ngdcsbld 1024 Nov 15 09:42 oprofile.out.RGR_ACR.1498048828
drwxr-xr-x 3 ngdcsbld ngdcsbld 1024 Nov 15 09:42 oprofile.out.RGR_ACR.1498049072
drwxr-xr-x 3 ngdcsbld ngdcsbld 1024 Nov 15 09:42 oprofile.out.RGR_AXF.1498047311
drwxr-xr-x 3 ngdcsbld ngdcsbld 1024 Nov 15 09:42 oprofile.out.RGR_AXF.1498047420
drwxr-xr-x 3 ngdcsbld ngdcsbld 1024 Nov 15 09:42 oprofile.out.RGR_AXF.1498047474
drwxr-xr-x 3 ngdcsbld ngdcsbld 1024 Nov 15 09:43 oprofile.out.RGR_AXF.1498047579
drwxr-xr-x 3 ngdcsbld ngdcsbld 1024 Nov 15 09:43 oprofile.out.RGR_AXF.1498048828
drwxr-xr-x 3 ngdcsbld ngdcsbld 1024 Nov 15 09:43 oprofile.out.RGR_BAG.1498047311
drwxr-xr-x 3 ngdcsbld ngdcsbld 1024 Nov 15 09:43 oprofile.out.RGR_BAG.1498047420
drwxr-xr-x 3 ngdcsbld ngdcsbld 1024 Nov 15 09:43 oprofile.out.RGR_BAG.1498047474
drwxr-xr-x 3 ngdcsbld ngdcsbld 1024 Nov 15 09:43 oprofile.out.RGR_BAG.1498047579
drwxr-xr-x 3 ngdcsbld ngdcsbld 1024 Nov 15 09:44 oprofile.out.RGR_BAG.1498048828
drwxr-xr-x 3 ngdcsbld ngdcsbld 1024 Nov 15 09:44 oprofile.out.RGR_BDQ.1498047420
drwxr-xr-x 3 ngdcsbld ngdcsbld 1024 Nov 15 09:44 oprofile.out.RGR_BRD.1498047579
drwxr-xr-x 3 ngdcsbld ngdcsbld 1024 Nov 15 09:44 oprofile.out.RGR_CPO.1498047311
drwxr-xr-x 3 ngdcsbld ngdcsbld 1024 Nov 15 09:44 oprofile.out.RGR_CPO.1498047420
drwxr-xr-x 3 ngdcsbld ngdcsbld 1024 Nov 15 09:44 oprofile.out.RGR_CPO.1498047474
drwxr-xr-x 3 ngdcsbld ngdcsbld 1024 Nov 15 09:45 oprofile.out.RGR_CPO.1498047508
drwxr-xr-x 3 ngdcsbld ngdcsbld 1024 Nov 15 09:45 oprofile.out.RGR_CPO.1498047579
drwxr-xr-x 3 ngdcsbld ngdcsbld 1024 Nov 15 09:45 oprofile.out.RGR_CPO.1498048828
drwxr-xr-x 3 ngdcsbld ngdcsbld 1024 Nov 15 09:45 oprofile.out.RGR_CPO.1498049072
drwxr-xr-x 3 ngdcsbld ngdcsbld 1024 Nov 15 09:45 oprofile.out.RGR_CPQ.1498049072
drwxr-xr-x 3 ngdcsbld ngdcsbld 1024 Nov 15 09:45 oprofile.out.RGR_CPR.1498047311
drwxr-xr-x 3 ngdcsbld ngdcsbld 1024 Nov 15 09:45 oprofile.out.RGR_CPR.1498047420
drwxr-xr-x 3 ngdcsbld ngdcsbld 1024 Nov 15 09:45 oprofile.out.RGR_CPR.1498047474
drwxr-xr-x 3 ngdcsbld ngdcsbld 1024 Nov 15 09:46 oprofile.out.RGR_CPR.1498047508
drwxr-xr-x 3 ngdcsbld ngdcsbld 1024 Nov 15 09:46 oprofile.out.RGR_CPR.1498047579
drwxr-xr-x 3 ngdcsbld ngdcsbld 1024 Nov 15 09:46 oprofile.out.RGR_CPR.1498049072
drwxr-xr-x 3 ngdcsbld ngdcsbld 1024 Nov 15 09:46 oprofile.out.RGR_CTF.1498047474
drwxr-xr-x 3 ngdcsbld ngdcsbld 1024 Nov 15 09:46 oprofile.out.RGR_CTF.1498047579
drwxr-xr-x 3 ngdcsbld ngdcsbld 1024 Nov 15 09:47 oprofile.out.RGR_CTF.1498048828
drwxr-xr-x 3 ngdcsbld ngdcsbld 1024 Nov 15 09:47 oprofile.out.RGR_EMD.1498047474

```

Ilustración 44. Ejemplo del los oprofile generados en cada subset de regresiones

Si por ejemplo lanzamos el comando siguiente podemos ver el arbol donde se encuentran los escenarios de por ejemplo el oprofile del subset de regresión RGR_FLI con id 1498047420

```

ngdcsbld@lonlnx48:~/operf/PRD_2017Q3+55-0-
83/oprofile.out.RGR_FLI.1498047420/samples> tree -L 17

```

```

34 directories, 0 files
ngdcsbld@lonlnx48:~/operf/PRD_2017Q3+55-0-83/oprofile.out.RGR_FLI.1498047420/samples> tree -L 17
.
`-- current
    |-- stats
    |   '-- throttled
    '-- {root}
        '-- data
            '-- mwdeldev
                '-- ngdpack-14
                    '-- mdw
                        '-- Pack
                            '-- 14-0-15-6
                                '-- bin
                                    '-- x86_64-2.6.16-v1
                                        '-- Release
                                            '-- otf_be
                                                '-- {dep}
                                                    |-- {anon:[vdso]}
                                                    |   '-- 6365.0x7ffff4a1ff000.0x7ffff4a1fffff
                                                    |       '-- {cg}
                                                    |           '-- {root}
                                                    '-- {anon:[vsyscall]}
                                                        '-- 6365.0xffffffffffff600000.0xffffffffffff600fff
                                                            '-- {cg}
                                                                '-- {root}
                                                                '-- {kern}
                                                                    '-- aacraid
                                                                        '-- {cg}
                                                                            '-- {kern}
                                                                    '-- bonding
                                                                        '-- {cg}
                                                                            '-- {kern}
                                                                            '-- {root}
                                                                    '-- ext3
                                                                        '-- {cg}
                                                                            '-- {kern}
                                                                            '-- {root}
                                                                    '-- forcedeth
                                                                        '-- {cg}
                                                                            '-- {kern}
                                                                            '-- {root}
                                                                    '-- ip_tables
                                                                        '-- {cg}
                                                                            '-- {root}
                                                                    '-- iptable_filter
                                                                    '-- jbd
                                                                        '-- {cg}
                                                                            '-- {kern}
                                                                            '-- {root}
                                                                    '-- nfs
                                                                        '-- {cg}
                                                                            '-- {kern}
                                                                            '-- {root}
                                                                    '-- scsi_mod
                                                                        '-- {cg}
                                                                            '-- {kern}
                                                                            '-- {root}
                                                                '-- sunrpc
                                                                    '-- {cg}

```

Ilustración 45. Arbol del subset de regresión RGR_FLI

A continuación se ha de ir hacia el directorio ~ngdcsbld/queues/operf/HISTORY como se muestra en las siguientes capturas de pantalla.

```

loninx48.lon.amadeus.net - PuTTY [vantiago@loninx48: ~]
...(>ngdcsbld/queues/opref/HISTORY)
`--> 11
total 3040
-rw-r--r-- 1 ngdcsbld ngdcsbld 25959 Apr 25 10:29 DETAIL.2018-04-25-09:26:02-UTC.tst:PRD_2018Q3:59-0-41:opref:::ncmde103 dbl.ok
-rw-r--r-- 1 ngdcsbld ngdcsbld 47196 Apr 25 10:39 DETAIL.2018-04-25-09:26:02-UTC.tst:PRD_2018Q3:59-0-41:opref:cmtec-ngxdcrgr:e2e.ncmde101 dbl.0k
-rw-r--r-- 1 ngdcsbld ngdcsbld 46885 Apr 25 10:45 DETAIL.2018-04-25-09:26:03-UTC.tst:PRD_2018Q3:59-0-41:opref:cmtec-ngxdcrgr:generated.ncmde101 dbl.0k.failed
-rw-r--r-- 1 ngdcsbld ngdcsbld 25733 Apr 25 10:37 DETAIL.2018-04-25-09:27:02-UTC.tst:PRD_2018Q3:59-0-41:opref:cmtec-ngrstrgr:livestore_cpr.ncmde102 dbl.0k.failed
-rw-r--r-- 1 ngdcsbld ngdcsbld 60672 Apr 25 10:38 DETAIL.2018-04-25-09:27:03-UTC.tst:PRD_2018Q3:59-0-41:opref:cmtec-ngrstrgr:livestore_cpr.misc.ncmde111 dbl.0k.ok
-rw-r--r-- 1 ngdcsbld ngdcsbld 74838 Apr 25 10:46 DETAIL.2018-04-25-09:27:05-UTC.tst:PRD_2018Q3:59-0-41:opref:cmtec-ngrlprgr:f11.ncmde102 dbl.0k
-rw-r--r-- 1 ngdcsbld ngdcsbld 40055 Apr 25 10:43 DETAIL.2018-04-25-09:27:06-UTC.tst:PRD_2018Q3:59-0-41:opref:cmtec-ngrlprgr:f11.ncmde103 dbl.0k
-rw-r--r-- 1 ngdcsbld ngdcsbld 75597 Apr 25 10:41 DETAIL.2018-04-25-09:27:08-UTC.tst:PRD_2018Q3:59-0-41:opref:cmtec-ngrlprgr:generated.ncmde103 dbl.0k
-rw-r--r-- 1 ngdcsbld ngdcsbld 43215 Apr 25 11:05 DETAIL.2018-04-25-09:27:04-UTC.tst:PRD_2018Q3:59-0-41:opref:cmtec-ngrlprgr:cpt.pushmax.ncmde102 dbl.0k
-rw-r--r-- 1 ngdcsbld ngdcsbld 58094 Apr 25 10:38 DETAIL.2018-04-25-09:31:02-UTC.tst:PRD_2018Q3:59-0-41:opref:cmtec-ngrlprgr:cpt.pushmax.ncmde103 dbl.0k
-rw-r--r-- 1 ngdcsbld ngdcsbld 47848 Apr 25 10:38 DETAIL.2018-04-25-09:31:02-UTC.tst:PRD_2018Q3:59-0-41:opref:cmtec-ngrlprgr:cpt.pushmax.ncmde103 dbl.ok
-rw-r--r-- 1 ngdcsbld ngdcsbld 45620 Apr 25 10:44 DETAIL.2018-04-25-09:37:02-UTC.tst:PRD_2018Q3:59-0-41:opref:cmapp-ngrlcrgr:mig.ncmde111 dbl.0k
-rw-r--r-- 1 ngdcsbld ngdcsbld 69959 Apr 25 11:10 DETAIL.2018-04-25-09:38:03-UTC.tst:PRD_2018Q3:59-0-41:opref:cmapp-ngrlcrgr:merge.ncmde102 dbl.0k
-rw-r--r-- 1 ngdcsbld ngdcsbld 92867 Apr 25 11:01 DETAIL.2018-04-25-09:39:02-UTC.tst:PRD_2018Q3:59-0-41:opref:cmapp-ngrlcrgr:ed17.ncmde102 dbl.0k
-rw-r--r-- 1 ngdcsbld ngdcsbld 131736 Apr 25 11:11 DETAIL.2018-04-25-09:39:03-UTC.tst:PRD_2018Q3:59-0-41:opref:cmapp-ngrlcrgr:ed16.ncmde103 dbl.0k
-rw-r--r-- 1 ngdcsbld ngdcsbld 130800 Apr 25 11:37 DETAIL.2018-04-25-09:40:04-UTC.tst:PRD_2018Q3:59-0-41:opref:cmapp-ngrlcrgr:ed15.ncmde101 dbl.0k
-rw-r--r-- 1 ngdcsbld ngdcsbld 130762 Apr 25 11:15 DETAIL.2018-04-25-09:42:03-UTC.tst:PRD_2018Q3:59-0-41:opref:cmapp-ngrlcrgr:ed14.ncmde103 dbl.0k
-rw-r--r-- 1 ngdcsbld ngdcsbld 130553 Apr 25 11:35 DETAIL.2018-04-25-09:43:02-UTC.tst:PRD_2018Q3:59-0-41:opref:cmapp-ngrlcrgr:ed13.ncmde102 dbl.0k
-rw-r--r-- 1 ngdcsbld ngdcsbld 161230 Apr 25 11:27 DETAIL.2018-04-25-09:45:01-UTC.tst:PRD_2018Q3:59-0-41:opref:cmapp-ngrlcrgr:ed11.ncmde111 dbl.0k
-rw-r--r-- 1 ngdcsbld ngdcsbld 130526 Apr 25 11:26 DETAIL.2018-04-25-09:45:02-UTC.tst:PRD_2018Q3:59-0-41:opref:cmapp-ngrlcrgr:ed10.ncmde102 dbl.0k
-rw-r--r-- 1 ngdcsbld ngdcsbld 43033 Apr 25 11:13 DETAIL.2018-04-25-09:45:05-UTC.tst:PRD_2018Q3:59-0-41:opref:cmapp-ngrlcrgr:cpt.pushmax.ncmde102 dbl.0k
-rw-r--r-- 1 ngdcsbld ngdcsbld 73860 Apr 25 11:31 DETAIL.2018-04-25-10:11:03-UTC.tst:PRD_2018Q3:59-0-41:opref:cmapp-ngrlcrgr:cpt.pushmax.ncmde102 dbl.0k
-rw-r--r-- 1 ngdcsbld ngdcsbld 7100 Apr 25 15:48 DETAIL.2018-04-25-14:48:02-UTC.gather_opref_data: vtmp_ngdcsbld_PRD_2018Q3:59-0-41_internal_cmtec_ngrstrgr_livestore_cpr.ncmde102 dbl.0k
-rw-r--r-- 1 ngdcsbld ngdcsbld 1831 Apr 25 15:48 DETAIL.2018-04-25-14:48:03-UTC.gather_opref_data: vtmp_ngdcsbld_PRD_2018Q3:59-0-41_internal_cmtec_ngrstrgr_livestore_cpr.misc.ncmde111 dbl.0k
-rw-r--r-- 1 ngdcsbld ngdcsbld 1862 Apr 25 15:48 DETAIL.2018-04-25-14:48:03-UTC.gather_opref_data: vtmp_ngdcsbld_PRD_2018Q3:59-0-41_internal_cmtec_ngxdcrgr_e2e.ncmde101 dbl.0k
-rw-r--r-- 1 ngdcsbld ngdcsbld 1796 Apr 25 15:53 DETAIL.2018-04-25-14:48:04-UTC.gather_opref_data: vtmp_ngdcsbld_PRD_2018Q3:59-0-41_internal_cmtec_nglxprgr_cpt.pushmax.ncmde101 dbl.0k
-rw-r--r-- 1 ngdcsbld ngdcsbld 1808 Apr 25 15:50 DETAIL.2018-04-25-14:48:04-UTC.gather_opref_data: vtmp_ngdcsbld_PRD_2018Q3:59-0-41_internal_cmtec_nglxprgr_cpt.pushmax.ncmde102 dbl.0k
-rw-r--r-- 1 ngdcsbld ngdcsbld 1760 Apr 25 15:54 DETAIL.2018-04-25-14:49:02-UTC.gather_opref_data: vtmp_ngdcsbld_PRD_2018Q3:59-0-41_internal_cmtec_nglxprgr_f11.ncmde103 dbl.0k
-rw-r--r-- 1 ngdcsbld ngdcsbld 1807 Apr 25 15:50 DETAIL.2018-04-25-14:49:02-UTC.gather_opref_data: vtmp_ngdcsbld_PRD_2018Q3:59-0-41_internal_cmtec_nglxprgr_f11.ncmde102 dbl.0k
-rw-r--r-- 1 ngdcsbld ngdcsbld 1772 Apr 25 15:52 DETAIL.2018-04-25-14:49:03-UTC.gather_opref_data: vtmp_ngdcsbld_PRD_2018Q3:59-0-41_internal_cmapp_ngrlcrgr_merge.ncmde102 dbl.0k
-rw-r--r-- 1 ngdcsbld ngdcsbld 1767 Apr 25 15:50 DETAIL.2018-04-25-14:49:03-UTC.gather_opref_data: vtmp_ngdcsbld_PRD_2018Q3:59-0-41_internal_cmapp_ngrlcrgr_mig.ncmde102 dbl.0k
-rw-r--r-- 1 ngdcsbld ngdcsbld 1766 Apr 25 15:49 DETAIL.2018-04-25-14:49:03-UTC.gather_opref_data: vtmp_ngdcsbld_PRD_2018Q3:59-0-41_internal_cmapp_ngrlcrgr_generated.ncmde101 dbl.0k
-rw-r--r-- 1 ngdcsbld ngdcsbld 1767 Apr 25 15:51 DETAIL.2018-04-25-14:51:03-UTC.gather_opref_data: vtmp_ngdcsbld_PRD_2018Q3:59-0-41_internal_cmapp_ngrlcrgr_ed14.ncmde102 dbl.0k
-rw-r--r-- 1 ngdcsbld ngdcsbld 1760 Apr 25 15:53 DETAIL.2018-04-25-14:51:03-UTC.gather_opref_data: vtmp_ngdcsbld_PRD_2018Q3:59-0-41_internal_cmapp_ngrlcrgr_ctp.ncmde102 dbl.0k
-rw-r--r-- 1 ngdcsbld ngdcsbld 1767 Apr 25 15:51 DETAIL.2018-04-25-14:51:03-UTC.gather_opref_data: vtmp_ngdcsbld_PRD_2018Q3:59-0-41_internal_cmapp_ngrlcrgr_ed14.ncmde102 dbl.0k
-rw-r--r-- 1 ngdcsbld ngdcsbld 1766 Apr 25 15:54 DETAIL.2018-04-25-14:51:03-UTC.gather_opref_data: vtmp_ngdcsbld_PRD_2018Q3:59-0-41_internal_cmapp_ngrlcrgr_ed15.ncmde102 dbl.0k
-rw-r--r-- 1 ngdcsbld ngdcsbld 1766 Apr 25 15:55 DETAIL.2018-04-25-14:50:07-UTC.gather_opref_data: vtmp_ngdcsbld_PRD_2018Q3:59-0-41_internal_cmapp_ngrlcrgr_ed16.ncmde102 dbl.0k
-rw-r--r-- 1 ngdcsbld ngdcsbld 1760 Apr 25 15:53 DETAIL.2018-04-25-14:51:03-UTC.gather_opref_data: vtmp_ngdcsbld_PRD_2018Q3:59-0-41_internal_cmapp_ngrlcrgr_ctp.ncmde102 dbl.0k
-rw-r--r-- 1 ngdcsbld ngdcsbld 1767 Apr 25 15:51 DETAIL.2018-04-25-14:51:03-UTC.gather_opref_data: vtmp_ngdcsbld_PRD_2018Q3:59-0-41_internal_cmapp_ngrlcrgr_ed17.ncmde101 dbl.0k
-rw-r--r-- 1 ngdcsbld ngdcsbld 1766 Apr 25 15:56 DETAIL.2018-04-25-14:52:03-UTC.gather_opref_data: vtmp_ngdcsbld_PRD_2018Q3:59-0-41_internal_cmapp_ngrlcrgr_ed11.ncmde102 dbl.0k
-rw-r--r-- 1 ngdcsbld ngdcsbld 1761 Apr 25 15:56 DETAIL.2018-04-25-14:53:02-UTC.gather_opref_data: vtmp_ngdcsbld_PRD_2018Q3:59-0-41_internal_cmapp_ngrlcrgr_cry.ncmde101 dbl.0k
-rw-r--r-- 1 ngdcsbld ngdcsbld 1767 Apr 25 15:55 DETAIL.2018-04-25-14:54:02-UTC.gather_opref_data: vtmp_ngdcsbld_PRD_2018Q3:59-0-41_internal_cmapp_ngrlcrgr_ed13.ncmde102 dbl.0k
-rw-r--r-- 1 ngdcsbld ngdcsbld 1767 Apr 25 15:58 DETAIL.2018-04-25-14:54:03-UTC.gather_opref_data: vtmp_ngdcsbld_PRD_2018Q3:59-0-41_internal_cmapp_ngrlcrgr_ed15.ncmde101 dbl.0k
-rw-r--r-- 1 ngdcsbld ngdcsbld 1797 Apr 25 15:56 DETAIL.2018-04-25-14:55:03-UTC.gather_opref_data: vtmp_ngdcsbld_PRD_2018Q3:59-0-41_internal_cmterc_ngrstrgr_generated.ncmde103 dbl.0k
-rw-r--r-- 1 ngdcsbld ngdcsbld 69610 Apr 26 12:51 DETAIL.2018-04-26-11:30:02-UTC.tst:PRD_2018Q3:59-0-41:opref:cmapp-ngrlcrgr:merge.ncmde102 dbl.0k
-rw-r--r-- 1 ngdcsbld ngdcsbld 7044 Apr 26 12:30 DETAIL.2018-04-26-11:30:02-UTC.tst:PRD_2018Q3:59-0-41:opref:cmapp-ngrlcrgr:mig.ncmde103 dbl.failed
-rw-r--r-- 1 ngdcsbld ngdcsbld 130279 Apr 26 12:21 DETAIL.2018-04-26-11:30:02-UTC.tst:PRD_2018Q3:59-0-41:opref:cmapp-ngrlcrgr:ctp.ncmde101 dbl.0k
-rw-r--r-- 1 ngdcsbld ngdcsbld 131351 Apr 26 12:17 DETAIL.2018-04-26-11:30:02-UTC.tst:PRD_2018Q3:59-0-41:opref:cmapp-ngrlcrgr:ed16.ncmde111 dbl.0k
-rw-r--r-- 1 ngdcsbld ngdcsbld 90469 Apr 26 12:52 DETAIL.2018-04-26-11:30:03-UTC.tst:PRD_2018Q3:59-0-44:opref:cmapp-ngrlcrgr:ed17.ncmde102 dbl.0k
-rw-r--r-- 1 ngdcsbld ngdcsbld 128666 Apr 26 13:16 DETAIL.2018-04-26-11:30:04-UTC.tst:PRD_2018Q3:59-0-44:opref:cmapp-ngrlcrgr:ed13.ncmde101 dbl.0k
-rw-r--r-- 1 ngdcsbld ngdcsbld 132063 Apr 26 13:22 DETAIL.2018-04-26-11:30:04-UTC.tst:PRD_2018Q3:59-0-44:opref:cmapp-ngrlcrgr:ed14.ncmde101 dbl.0k
-rw-r--r-- 1 ngdcsbld ngdcsbld 131596 Apr 26 13:24 DETAIL.2018-04-26-11:30:04-UTC.tst:PRD_2018Q3:59-0-44:opref:cmapp-ngrlcrgr:ed15.ncmde111 dbl.0k
-rw-r--r-- 1 ngdcsbld ngdcsbld 160706 Apr 26 13:24 DETAIL.2018-04-26-11:30:08-UTC.tst:PRD_2018Q3:59-0-44:opref:cmapp-ngrlcrgr:ed11.ncmde102 dbl.0k
-rw-r--r-- 1 ngdcsbld ngdcsbld 7044 Apr 26 12:31 DETAIL.2018-04-26-11:31:02-UTC.tst:PRD_2018Q3:59-0-44:opref:cmapp-ngrlcrgr:cry.ncmde103 dbl.failed
...(>ngdcsbld/queues/opref/HISTORY)
`--> [REDACTED] [vantiago@loninx48: ~]

```

```

loninx48.lon.amadeus.net - PuTTY [vantiago@loninx48: ~]
...(>ngdcsbld/queues/opref/HISTORY)
`--> 11
total 3040
-rw-r--r-- 1 ngdcsbld ngdcsbld 25959 Apr 25 10:29 DETAIL.2018-04-25-09:26:02-UTC.tst:PRD_2018Q3:59-0-41:opref:::ncmde103 dbl.ok
-rw-r--r-- 1 ngdcsbld ngdcsbld 47196 Apr 25 10:39 DETAIL.2018-04-25-09:26:02-UTC.tst:PRD_2018Q3:59-0-41:opref:cmtec-ngxdcrgr:e2e.ncmde101 dbl.0k
-rw-r--r-- 1 ngdcsbld ngdcsbld 46885 Apr 25 10:45 DETAIL.2018-04-25-09:26:03-UTC.tst:PRD_2018Q3:59-0-41:opref:cmtec-ngxdcrgr:generated.ncmde101 dbl.0k.failed
-rw-r--r-- 1 ngdcsbld ngdcsbld 25733 Apr 25 10:37 DETAIL.2018-04-25-09:27:02-UTC.tst:PRD_2018Q3:59-0-41:opref:cmtec-ngrstrgr:livestore_cpr.ncmde102 dbl.0k.failed
-rw-r--r-- 1 ngdcsbld ngdcsbld 60672 Apr 25 10:38 DETAIL.2018-04-25-09:27:02-UTC.tst:PRD_2018Q3:59-0-41:opref:cmtec-ngrstrgr:livestore_cpr.misc.ncmde111 dbl.0k.ok
-rw-r--r-- 1 ngdcsbld ngdcsbld 74383 Apr 25 10:43 DETAIL.2018-04-25-09:27:03-UTC.tst:PRD_2018Q3:59-0-41:opref:cmtec-ngrlprgr:f11.ncmde102 dbl.0k
-rw-r--r-- 1 ngdcsbld ngdcsbld 97095 Apr 25 12:13 DETAIL.2018-04-25-09:27:03-UTC.tst:PRD_2018Q3:59-0-41:opref:cmtec-ngrlprgr:generated.ncmde111 dbl.0k
-rw-r--r-- 1 ngdcsbld ngdcsbld 47194 Apr 25 10:30 DETAIL.2018-04-25-11:30:02-UTC.tst:PRD_2018Q3:59-0-41:opref:cmapp-ngrlcrgr:mig.ncmde103 dbl.failed
-rw-r--r-- 1 ngdcsbld ngdcsbld 130279 Apr 25 10:21 DETAIL.2018-04-26-11:30:02-UTC.tst:PRD_2018Q3:59-0-41:opref:cmapp-ngrlcrgr:ctp.ncmde101 dbl.0k
-rw-r--r-- 1 ngdcsbld ngdcsbld 131351 Apr 26 12:17 DETAIL.2018-04-26-11:30:02-UTC.tst:PRD_2018Q3:59-0-41:opref:cmapp-ngrlcrgr:ed16.ncmde111 dbl.0k
-rw-r--r-- 1 ngdcsbld ngdcsbld 90469 Apr 26 12:52 DETAIL.2018-04-26-11:30:03-UTC.tst:PRD_2018Q3:59-0-44:opref:cmapp-ngrlcrgr:ed17.ncmde102 dbl.0k
-rw-r--r-- 1 ngdcsbld ngdcsbld 128666 Apr 26 13:16 DETAIL.2018-04-26-11:30:04-UTC.tst:PRD_2018Q3:59-0-44:opref:cmapp-ngrlcrgr:ed13.ncmde101 dbl.0k
-rw-r--r-- 1 ngdcsbld ngdcsbld 132063 Apr 26 13:22 DETAIL.2018-04-26-11:30:04-UTC.tst:PRD_2018Q3:59-0-44:opref:cmapp-ngrlcrgr:ed14.ncmde101 dbl.0k
-rw-r--r-- 1 ngdcsbld ngdcsbld 131596 Apr 26 13:24 DETAIL.2018-04-26-11:30:04-UTC.tst:PRD_2018Q3:59-0-44:opref:cmapp-ngrlcrgr:ed15.ncmde111 dbl.0k
-rw-r--r-- 1 ngdcsbld ngdcsbld 160706 Apr 26 13:24 DETAIL.2018-04-26-11:30:08-UTC.tst:PRD_2018Q3:59-0-44:opref:cmapp-ngrlcrgr:ed11.ncmde102 dbl.0k
-rw-r--r-- 1 ngdcsbld ngdcsbld 7044 Apr 26 12:31 DETAIL.2018-04-26-11:31:02-UTC.tst:PRD_2018Q3:59-0-44:opref:cmapp-ngrlcrgr:cry.ncmde103 dbl.failed
...(>ngdcsbld/queues/opref/HISTORY)
`--> [REDACTED] [vantiago@loninx48: ~]

```

Ilustración 46. History del set de regresiones

Seguidamente se ha de abrir este fichero por ejemplo con vim para visualizar su contenido por ejemplo del uno de los generados en el cual podremos ver su id y los subsets que abarca ademas del resto de información general de este set.

vim DETAIL.2018-04-25-14:55:03-UTC.gather_opref_data: vtmp_ngdcsbld_PRD_2018Q3_59-0-41_internal_cmtec_ngrstrgr_generated.ncmde103_db1.ok

```

lonlnx48.lon.amadeus.net - PuTTY
1 TASKID gather_operf_data
2 REQUESTNAME gather_operf_data: vtmp_ngdcsbld_PRD_2018Q3_59-0-41_internal_cmtec_ngrstrgr_generated
3 PATH /vtmp/ngdcsbld/PRD_2018Q3/59-0-41/internal/cmtec/ngrstrgr/generated
4 HOST lonlnx34
5 DUMP_DIR PRD_2018Q3+59-0-41
6 ++ Wed 2018-04-25 14:55:04 UTC INFO user=ngdcsbld, host=lonlnx51, pid=18542
7 ++ Wed 2018-04-25 14:55:04 UTC INFO Task is '/data/ngdtools/py/cronbin/0-0-83/lib/python2.7/site-packages/cronbin/tasks/gather_operf_data.py'.
8 ++ Wed 2018-04-25 14:55:04 UTC INFO Port range for ncmde103_db1 ( 25401 : 25500 ) is free? True
9 ++ Wed 2018-04-25 14:55:04 UTC INFO Gathering operf data from '/vtmp/ngdcsbld/PRD_2018Q3/59-0-41/internal/cmtec/ngrstrgr/generated' to /projects/ngdcsbld/operf/PRD_2018Q3+59-0-41
10 ++ Wed 2018-04-25 14:55:04 UTC INFO About to execute command 'ssh lonlnx34 stat /vtmp/ngdcsbld/PRD_2018Q3/59-0-41/internal/cmtec/ngrstrgr/generated' to check source exists..
11   File: '/vtmp/ngdcsbld/PRD_2018Q3/59-0-41/internal/cmtec/ngrstrgr/generated'
12   Size: 4096          Blocks: 8          IO Block: 4096   directory
13  Device: 811h/2065d   Inode: 74171553   Links: 6
14  Access: (0775/drwxrwxr-x)  Uid: ( 4058/ngdcsbld)  Gid: ( 4012/ngdcsbld)
15  Access: 2018-04-25 10:04:55.000000000 +0000
16  Modify: 2018-04-25 11:13:31.000000000 +0000
17  Change: 2018-04-25 11:13:31.000000000 +0000
18 ++ Wed 2018-04-25 14:55:06 UTC INFO Action 'check source exists.' completed ok.
19 ++ Wed 2018-04-25 14:55:06 UTC INFO About to execute command 'rsync -az lonlnx34:/vtmp/ngdcsbld/PRD_2018Q3/59-0-41/internal/cmtec/ngrstrgr/generated' to collect operf data..
20 ++ Wed 2018-04-25 14:56:27 UTC INFO Action 'collect operf data.' completed ok.
21 ++ Wed 2018-04-25 14:56:27 UTC INFO Done.
22 ++ Wed 2018-04-25 14:56:27 UTC INFO -> task successful

lonlnx48.lon.amadeus.net - PuTTY
1 TASKID gather_operf_data
2 REQUESTNAME gather_operf_data: vtmp_ngdcsbld_PRD_2018Q3_59-0-41_internal_cmtec_ngrstrgr_generated
3 PATH /vtmp/ngdcsbld/PRD_2018Q3/59-0-41/internal/cmtec/ngrstrgr/generated
4 HOST lonlnx34
5 DUMP_DIR PRD_2018Q3+59-0-41
6 ++ Wed 2018-04-25 14:55:04 UTC INFO user=ngdcsbld, host=lonlnx51, pid=18542
7 ++ Wed 2018-04-25 14:55:04 UTC INFO Task is '/data/ngdtools/py/cronbin/0-0-83/lib/python2.7/site-packages/cronbin/tasks/gather_operf_data.py'.
8 ++ Wed 2018-04-25 14:55:04 UTC INFO Port range for ncmde103_db1 ( 25401 : 25500 ) is free? True
9 ++ Wed 2018-04-25 14:55:04 UTC INFO Gathering operf data from /vtmp/ngdcsbld/PRD_2018Q3/59-0-41/internal/cmtec/ngrstrgr/generated to /projects/ngdcsbld/operf/PRD_2018Q3+59-0-41
10 ++ Wed 2018-04-25 14:55:04 UTC INFO About to execute command 'ssh lonlnx34 stat /vtmp/ngdcsbld/PRD_2018Q3/59-0-41/internal/cmtec/ngrstrgr/generated' to check source exists..
11   File: '/vtmp/ngdcsbld/PRD_2018Q3/59-0-41/internal/cmtec/ngrstrgr/generated'
12   Size: 4096          Blocks: 8          IO Block: 4096   directory
13  Device: 811h/2065d   Inode: 74171553   Links: 6
14  Access: (0775/drwxrwxr-x)  Uid: ( 4058/ngdcsbld)  Gid: ( 4012/ngdcsbld)
15  Access: 2018-04-25 10:04:55.000000000 +0000
16  Modify: 2018-04-25 11:13:31.000000000 +0000
17  Change: 2018-04-25 11:13:31.000000000 +0000
18 ++ Wed 2018-04-25 14:55:06 UTC INFO Action 'check source exists.' completed ok.
19 ++ Wed 2018-04-25 14:55:06 UTC INFO About to execute command 'rsync -az lonlnx34:/vtmp/ngdcsbld/PRD_2018Q3/59-0-41/internal/cmtec/ngrstrgr/generated' to collect operf data..
20 ++ Wed 2018-04-25 14:56:27 UTC INFO Action 'collect operf data.' completed ok.
21 ++ Wed 2018-04-25 14:56:27 UTC INFO Done.
22 ++ Wed 2018-04-25 14:56:27 UTC INFO -> task successful

<anal_cmtec_ngrstrgr_generated.ncmde103_db1.ok [RO] 20,78 All>

```

Ilustración 47. Contenido del subset de regresión y enlace al fichero generado.

5.2. MÓDULO 1: Generación del JSON referente a cada subset de regresiones por funciones.

Se ha de comentar que el módulo 1 depende en una gran medida del módulo 0 ya que el principal propósito de este módulo consiste en generar un fichero JSON en el cual se almacenan por cada subset de regresiones los ficheros csv los cuales contienen el identificador de subset de regresión y las funciones que testean estos subsets.

Para ello se han generado los correspondientes ficheros csv por cada oprofile. Estos vienen a ser 3 ficheros: names, functions, relations.

A continuación se muestra una captura de uno del contenido de uno de los ficheros csv donde se pueden apreciar los nombres de las funciones que testeó ese subset de regresiones en concreto.

	:funcHash:ID	name	lib	:LABEL
1	0000b48eac55abb699d1b	std::pair<BMT::BagMessagesEventTriggersBzrCache::BzRuleSearchCriteria const, std::vector<BMT::BagMessagesEventTriggersBzrCache::BzRuleSearchCriteria>>	/remote/vtmp2/ngdcs.jobrep/cmbmt/ngbmt/lib/x	Function
2	0001e973ee9f64b5b8f7f7	std::Rb_tree<COM::FldUniqueFlightLegIdentifier, std::pair<COM::FldUnic/remote/vtmp2/ngdcs.jobrep/cmbmt/ngbmt/lib/x	Function	
3	00022bd41f3b687dc9347a	std::Select1st<std::pair<unsigned int const, avlmc::EquipmentRecord*>>::op	/data/mwdeldev/ngdp/14/pfx/AVLMCT/16-140-	Function
4	0002db7f39f7e5778f742f3	std::Rb_tree<COM::BasObservable*, std::pair<COM::BasObservable* const, dt::remote/vtmp2/ngdcs.jobrep/cmbmt/ngbmt/lib/x	Function	
5	00052e833532b6c1793748	CMG_Conversation::setLastRcvdSeqNumber(unsigned int)	/data/mwdeldev/ngdp/14/mdw/Pack/14-0-15-2	Function
6	0008034ce0d8ab9e4e7d0	COM::BasTransactionAbstract::RegisterListener(dbo::TransactionListener&)	/data/mwdeldev/ngdp/14/ngd/core/ngdbo/14-	Function
7	0009e7346a50a659ec2572	COM::FldStringAbstract::FldStringAbstract(int)	/remote/vtmp2/ngdcs.jobrep/cmapp/ngsyn/lib/x8	Function
8	000d13b8ad48b6ef033d3	std::Rb_tree<ABR::RUL::FldCriterionCode, std::pair<ABR::RUL::FldCriterionCoc::data, mwdeldev/ngdp/14/ngd/ngmds/14-0-14-	Function	
9	000df1d6504291456776cc	std::Rb_tree<COM::FldTransactionId, std::pair<COM::FldTransactionId const, s::remote/vtmp2/ngdcs.jobrep/cmapq/ngacs/lib/x8	Function	
10	000e8d22504b53c624b382	(anonymous namespace)::cleanConnectionPoolIntervalExpired()	/data/mwdeldev/ngdp/14/ngd/core/ngdbo/14-	Function
11	000ea5619ae964babd434	std::Vector_base<ABR::DCS::BinIPS_0_1_Rule, std::allocator<ABR::DCS::BinIPS_0_1_Rule>::alloc	/data/mwdeldev/ngdp/14/ngd/ngmds/14-0-14-	Function
12	000e6f0a2687fe6c41b8l	std::map<char const*, bool (DbUpVector<TFmFlightLegExcessBag::>::operator()>(unsigned int))	/data/vtmp/ngrpmlb/RebuildLocalRepository/bm	Function
13	000f0f83fde4991035d753	std::deque<std::pair<char const*, bool (DbUpVector<TFmDeadload::>::operator()>(unsigned int))>::operator()	/data/vtmp/ngrpmlb/RebuildLocalRepository/bm	Function
14	000f42aa06ad77d830e0f9	SMT::EdISSR_Refresh_11::~EdISSR_Refresh_11()	/remote/vtmp2/ngdcs.jobrep/cmapp/ngsmt/lib/x8	Function
15	001116ab944da595f81746	std::Vector_base<ABR::DCS::BinSPB_0_1_Rule, std::allocator<ABR::DCS::BinSPB_0_1_Rule>::alloc	/data/mwdeldev/ngdp/14/ngd/ngmds/14-0-14-	Function
16	0011c5537ea4bf9b84fd6	boost::arg<x>::arg()	/remote/vtmp2/ngdcs.jobrep/cmapp/ngsec/lib/x8	Function
17	00143542763994987db3f3	void boost::detail::sp_pointer_construct<CMT::CatGenLegacyProductFacetFact>::operator()()	/remote/vtmp2/ngdcs.jobrep/cmapp/ngcmt/lib/x8	Function
18	001452814a1b3c63d3acd7	std::map<COM::FldBzrIdentifier, COM::BzRuleAbstract*, std::less<COM::FldBzr/remote/vtmp2/ngdcs.jobrep/cmapp/ngcmt/lib/x8	Function	
19	00154c1bf396b9b83704e4	boost::unordered::detail::table<boost::unordered::detail::table<boost::unordered::detail::set<std::allocator<st::remote/mw/deliveries/DcsDecodingAPI/2-13/lib/	Function	
20	00156c25c158373a18fb61!	CMT::FldCappsStatus()	/remote/vtmp2/ngdcs.jobrep/cmapp/ngcmt/lib/x8	Function
21	0016d291a29ff88e2d523	BMT::Bag::getNotDepartedBagLegDeliveriesFromPort(std::vector<BMT::BagLeg>)	/remote/vtmp2/ngdcs.jobrep/cmbmt/ngbmt/lib/x	Function
22	0016d651b2fc2c2732a094	std::Rb_tree<char const*, std::pair<char const* const, void (ABR::DCS::BinSCE_>::operator()>)	/data/mwdeldev/ngdp/14/ngd/ngmds/14-0-14-	Function

Ilustración 48. Ejemplo de fichero de funciones csv.

Para su visualización y mejor entendimiento de forma general se ha usado Neo4J en el cual nos ha permitido comprender de mejor forma las relaciones entre las distintas funciones que se hayan en cada uno de los test de regresión.

Seguidamente se ha escrito el fichero db1.py el cual nos ha generado el correspondiente fichero json formado por todos los nombres de los ficheros csv como claves y las funciones contenidas en estos ficheros como valores.

El nombre del fichero csv se corresponde con el path en el que se haya esa subregresión con su correspondiente ID identificativo.

El procedimiento seguido por este módulo resulta ser el que viene mostrado por la siguiente ilustración donde se obtienen los ficheros csv donde se coge del fichero func.csv los nombres de las funciones que se hayan implicadas en los subsets de regresión concretos.

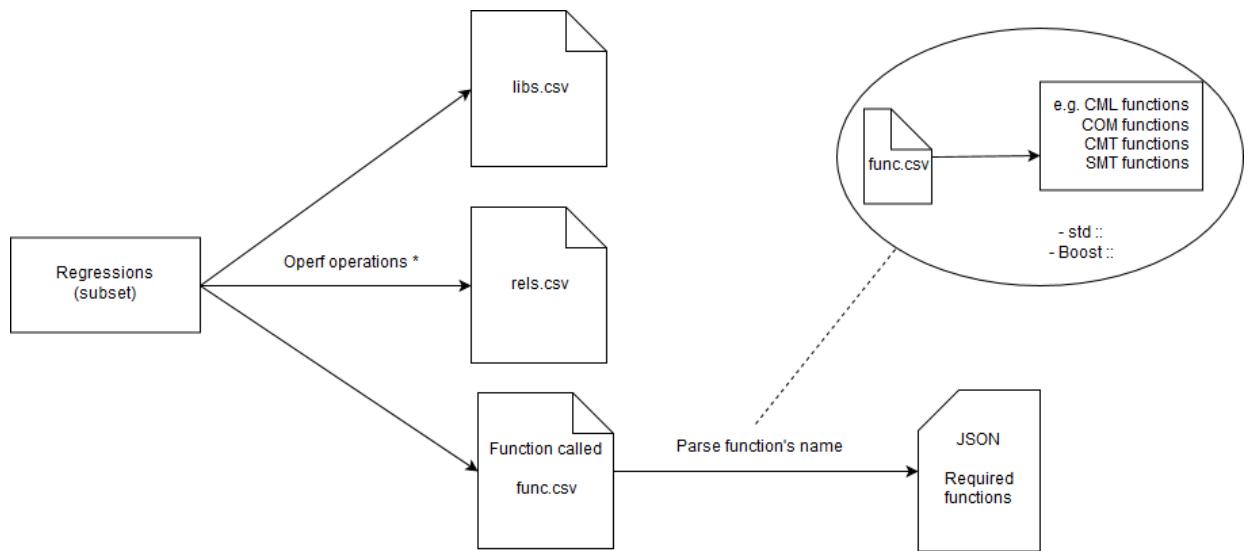


Ilustración 49. Procedimiento del módulo 1

db1.py - Z:\task5_vtmp\db1-RegFunc -- Atom

File Edit View Selection Find Packages Help

Project

memory.txt gather_opref_data.py generate_opref_files.py make_csvs.py db1.py parser_db2.py db3_reg_to_execute.py

Documents

Dell Downloads

My Music

My Pictures

My Videos

MyjabberChats

Myjabberfiles

OneNote Notebooks

VictoriaDocs

Virtual Machines

desktop.ini

memory.txt

```
1 import csv
2 import json
3 import glob
4
5 #path to csv files
6 PATH = '/remote/users/vsantago/task5_vtmp/callgraph_to_csv/oreport2neo4j/func_csv_files'
7 IGNORED_FUNC = ['std::', 'boost::']
8
9 def retrieve_reg_cpp_functions_db(PATH):
10     """
11         Retrieves a json file where the keys are the path to the regressions and
12         the values are the functions declarations.
13     """
14     :PATH this is the path to the csv files
15     """
16     dic_1 = {}
17     for csv_file in glob.iglob(PATH+'func_oprofile.out.*'):
18         print csv_file
19         dic_1[csv_file] = get_list_func_names(csv_file)
20     get_json_file(dic_1)
21
22
23 def get_list_func_names(csv_file):
24     """
25         Retrieves one list with all the names per csv file
26         :csv_file, path to csv files
27
28         :return names_list (list)
29     """
30     info = []
31     info_list = []
32     names_list = []
33     with open(csv_file) as func_file:
34         readfile = csv.reader(func_file, delimiter=' ', quotechar='\'')
35         for row in readfile:
36             info.append(''.join(row))
37         for each_row in info:
38             info_list.append(each_row.split(','))
39         for line in info_list[1:]:
40             line.append(line[2].split(',')) 
41             line.pop(2)
42             names_list.append(line[1])
```

Ilustración 50. Código de la herramienta del Módulo 1.

```

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
 regressions_db.txt elasticsearch_kibana.sh new 1 make_csv.py UbuntuRegressions.vmx db1_Reg_FNames.json db2_Cpp_FNames.json db3_regressions_to_execute_old.json aaaaabcccc.txt db3_regressions_to_e
1 ["remote/users/vsantiago/task5_vtmp/callgraph_to_csv/opreport2neo4j/func_csv_files/func_oprofile.out.RGR_LVP.1524649069_call_graph.csv";
2 ["apic_timer_interrupt\"", "ret_from_intr\"", "page_fault\"", "retint_swaps\"", "account_process_tick\"", "error_sti\"", "sqlite3VdbeExec\"", "acct_update_integr
3 "/remote/users/vsantiago/task5_vtmp/callgraph_to_csv/opreport2neo4j/func_csv_files/func_oprofile.out.RGR_IDC.1524648912_call_graph.csv";
4 ["ip_local_deliver_finish\"", "ret_from_intr\"", "apic_timer_interrupt\"", "page_fault\"", "call_function_single_interrupt\"", "error_entry\"", "irq_entries_start
5 ["/remote/users/vsantiago/task5_vtmp/callgraph_to_csv/opreport2neo4j/func_csv_files/func_oprofile.out.RGR_LVP.1524649069_call_graph.csv": {"apic_timer_interrupt"
6 ]

```

Ilustración 51. Ejemplo del fichero JSON generado.

Ficero JSON generado en el cual se contienen como claves las direcciones a los distintos ficheros csv donde se hayan especificados los paths a los subsets de regresiones declarados y diferenciados además con sus Ids lo cual los hace únicamente diferenciables.

La siguiente imagen muestra un ejemplo de ejecución la cual ha generado la correspondiente información en el fichero json de los ficheros csv que se muestran:

```

--> ls
CppHeaderParser-2.7.4 TFM_Reg_Project.zip dbl-RegFunc db2_example.py dbnodes.py new_db2_parser.py oprofile.out.RGR_RST.1524650635_call_graph
CppHeaderParser.py callgraph_to_csv db2-CppFunc db3-MapReg gather_operf_data.py oprofile.out.RGR_IDC.1524648912_call_graph parser.py
Makefile clang-master db2.py db1_json make_opreport oprofile.out.RGR_LVP.1524649069_call_graph tasks
--> cd dbl-RegFunc
--> ./task5_vtmp
--> cd dbl-RegFunc
--> ./task5_vtmp/dbl-RegFunc
--> ls
dbl.py dbl_Reg_FNames.json
--> ./task5_vtmp/dbl-RegFunc
--> python dbl.py
/remote/users/vsantiago/task5_vtmp/callgraph_to_csv/opreport2neo4j/func_csv_files/func_oprofile.out.RGR_IDC.1524648912_call_graph.csv
/remote/users/vsantiago/task5_vtmp/callgraph_to_csv/opreport2neo4j/func_csv_files/func_oprofile.out.RGR_LVP.1524649069_call_graph.csv
--> ./task5_vtmp/dbl-RegFunc
--> ./task5_vtmp/dbl-RegFunc
--> █

```

Ilustración 52. Ejemplo del Módulo 1

En la siguiente imagen se tiene otro ejemplo del fichero JSON obtenido en este módulo 1 en el cual se puede apreciar que la claves es como se ha visto en la imagen anterior, el path hacia donde se encuentra el subset de regesion y su id y como valores las funciones que tiene.

Ilustración 53. Fichero JSON obtenido del módulo 1.

Con ello se tendría terminado y definido el funcionamiento principal de este módulo de nuestra herramienta.

5.3. MÓDULO 2: Parseando el código C++ con Clang

Este módulo comprende las actividades que se realizan sobre los códigos c++ del producto de la compañía, los cuales necesitan ser testeados por los desarrolladores mediante las pruebas de regresión.

El procedimiento que sigue es relativamente sencillo de entender puesto que en este módulo las acciones que se realizan consisten básicamente en obtener los ficheros que se han modificado como si fuese un diff y obtener de estos las funciones que contienen con el fin de almacenarlas en una base de datos o fichero JSON el cual será accedido por el módulo 3.

A grandes rasgos el procedimiento resulta parecido al descrito por la siguiente ilustración:

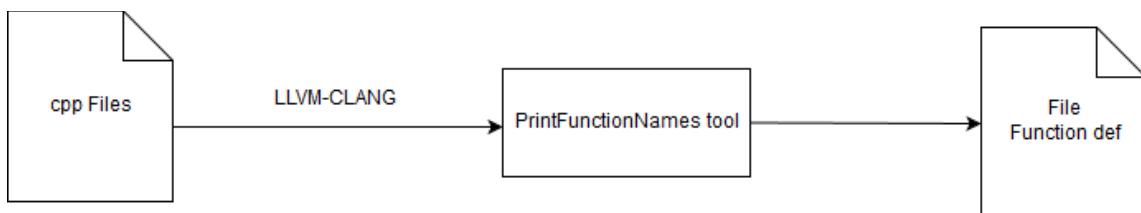


Ilustración 54. Procedimiento seguido por el módulo 2.

Tras optar por usar la herramienta Clang, se inició instalándola en la máquina de trabajo actual (lonlnx48).

Los pasos seguidos para su uso son simplemente importar la biblioteca y establecer el correspondiente path. Queda como sigue abriendo por ejemplo ipython:

- import clang
- import clang.cindex
- clang.cindex.Config.set_library_path('/opt/1A/toolchain/x86_64-2.6.32-v4.0.11/lib')
- idx = clang.cindex.Index.create()

El error más común que se obtiene es el siguiente:

ERROR: /lib64/libc.so.6: version `GLIBC_2.14' not found (required by /opt/1A/toolchain/x86_64-2.6.32-v2.0.17/lib/libclang.so). To provide a path to libclang use Config.set_library_path() or Config.set_library_file().

```

Ioninx48.lion.amadeus.net - PUTTY
--> 198     value = self.wrapped(instance)
  199     setattr(instance, self.wrapped.__name__, value)
 200

\data/vtmp/shadow/ioninx48/remote/users/vsantiago/task5_vtmp/clang-master/bindings/python/clang/cindex.py in lib(self)
 4101     @CachedProperty
 4102     def lib(self):
--> 4103         lib = self.get_cindex_library()
 4104         register_functions(lib, not Config.compatibility_check)
 4105         Config.loaded = True

\data/vtmp/shadow/ioninx48/remote/users/vsantiago/task5_vtmp/clang-master/bindings/python/clang/cindex.py in get_cindex_library(self)
 4132             "Config.set_library_path() or \"\n 4133             \"Config.set_library_file()\""
--> 4134             raise LibclangError(msg)
 4135
 4136     return library

LibclangError: /lib64/libc.so.6: version `GLIBC_2.14' not found (required by /opt/la/toolchain/x86_64-2.6.32-v4.0.11/lib/libclang.so). To provide a path to libclang use Config.set_library_path() or Config.set_library_file().

In [45]: clang.cindex.Config.set_library_path('/opt/la/toolchain/x86_64-2.6.32-v2.0.17/lib')

In [46]: idx = clang.cindex.Index.create()
          -----
          LibclangError                                 Traceback (most recent call last)
<ipython-input-46-87211872b8a1> in <module>()
----> 1 idx = clang.cindex.Index.create()

\data/vtmp/shadow/ioninx48/remote/users/vsantiago/task5_vtmp/clang-master/bindings/python/clang/cindex.py in create(excludeDecls)
 2664     """
 2665     excludeDecls -- Exclude local declarations from translation units.
--> 2666     return Index(conf.lib clang_createIndex(excludeDecls, 0))
 2667
 2668     def __del__(self):

\data/vtmp/shadow/ioninx48/remote/users/vsantiago/task5_vtmp/clang-master/bindings/python/clang/cindex.py in __get__(self, instance, instance_type)
 196     return self
 197
--> 198     value = self.wrapped(instance)
 199     setattr(instance, self.wrapped.__name__, value)
 200

\data/vtmp/shadow/ioninx48/remote/users/vsantiago/task5_vtmp/clang-master/bindings/python/clang/cindex.py in lib(self)
 4101     @CachedProperty
 4102     def lib(self):
--> 4103         lib = self.get_cindex_library()
 4104         register_functions(lib, not Config.compatibility_check)
 4105         Config.loaded = True

\data/vtmp/shadow/ioninx48/remote/users/vsantiago/task5_vtmp/clang-master/bindings/python/clang/cindex.py in get_cindex_library(self)
 4132             "Config.set_library_path() or \"\n 4133             \"Config.set_library_file()\""
--> 4134             raise LibclangError(msg)
 4135
 4136     return library

LibclangError: /lib64/libc.so.6: version `GLIBC_2.14' not found (required by /opt/la/toolchain/x86_64-2.6.32-v2.0.17/lib/libclang.so). To provide a path to libclang use Config.set_library_path() or Config.set_library_file().

In [47]:

```

Ilustración 55. Error con Clang en ipython

La solución que se le dio fue exportar el path y su LD_LIBRARY_PATH como se puede ver en la primera página de anexos.

Seguidamente se encontró el problema de que por problemas internos de la configuración de la maquina puesto que se hallaba configurada para un uso de Clang distinto al que se le iba a dar en nuestro proyecto el cual no permitía la obtención de los resultados esperados por lo que se procedió a realizar este módulo en una maquina virtual.

- Creación de la máquina virtual para el uso de Clang:
 - En primer lugar se configuró un proxy nuevo con squid y se añadió a la conexión de nuestra maquina virtual.

Para ello se siguieron los siguientes pasos:

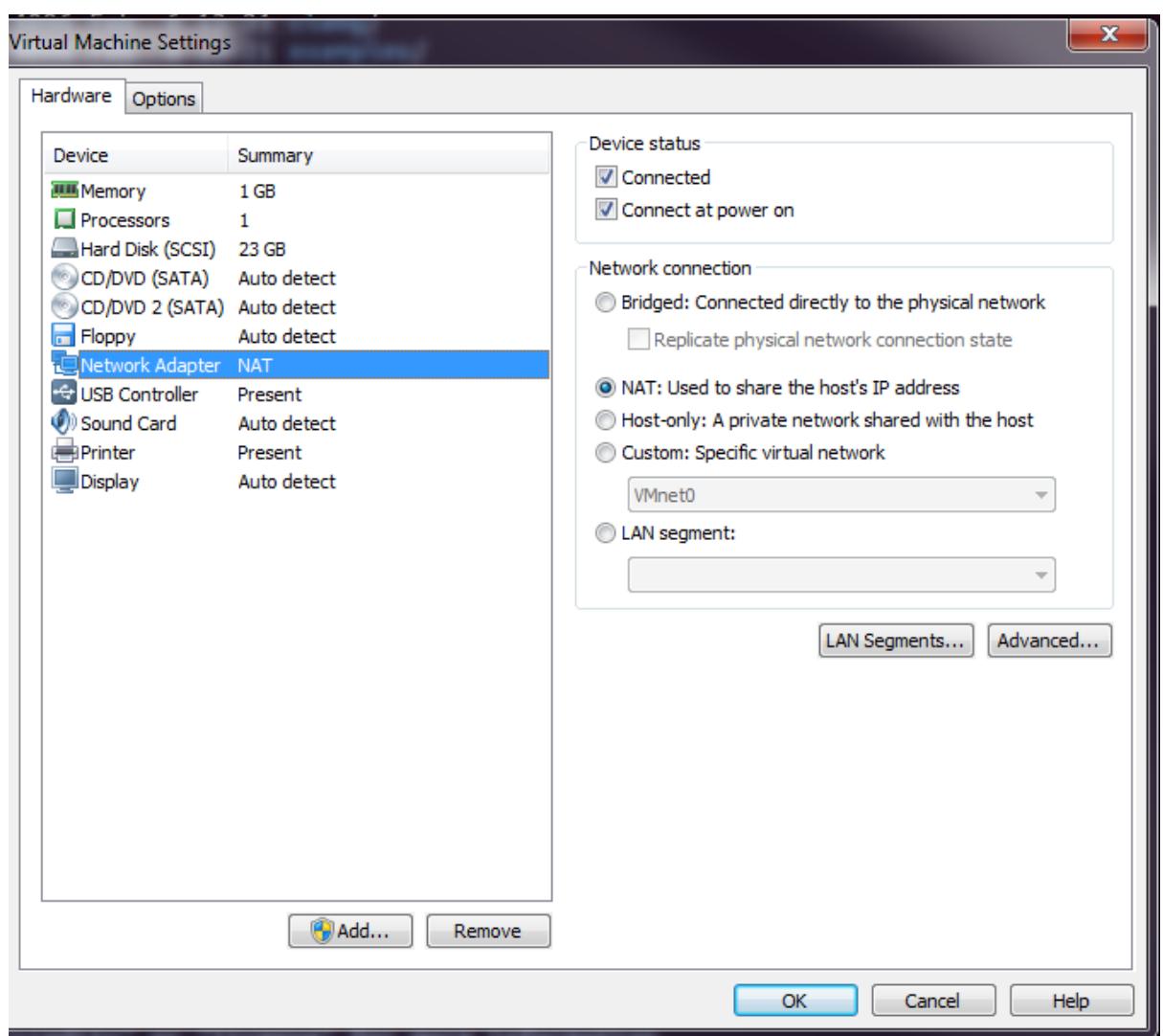
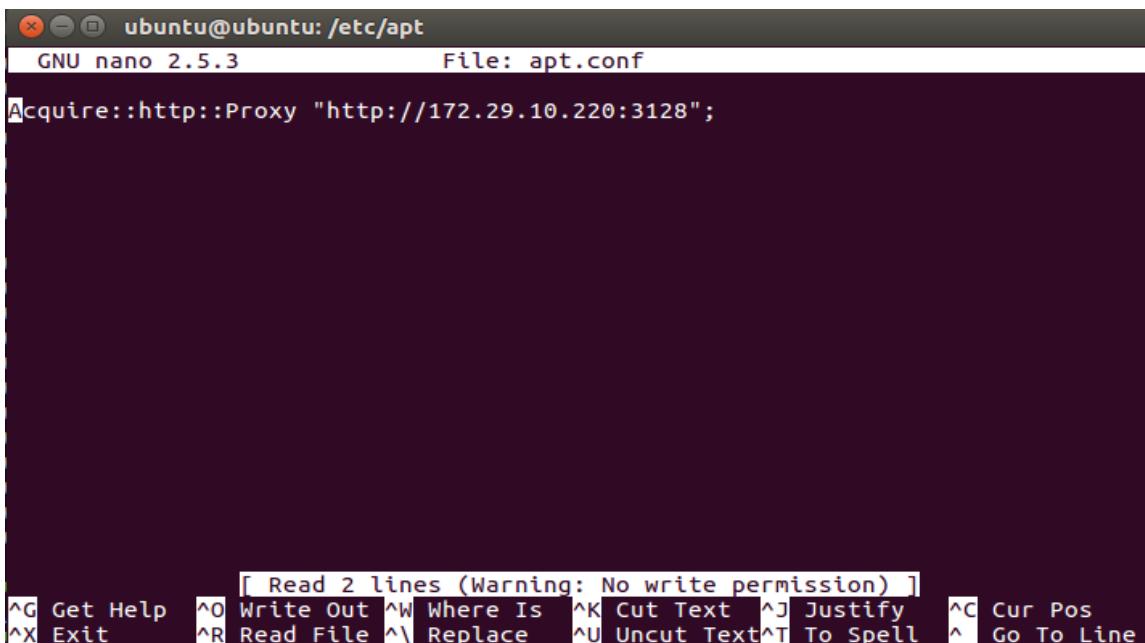


Ilustración 56. Configuración de la máquina virtual

- Exportación del proxy:
 - export
https_proxy=http://ncegcochecks.nce.amadeus.net:3128
 - export
http_proxy=http://ncegcochecks.nce.amadeus.net:3128
 - export
ftp_proxy=http://ncegcochecks.nce.amadeus.net:3128
- Crear el fichero: /etc/apt/apt.conf
- Insertar ->Acquire::http::Proxy "http://yourproxyaddress:proxyport";



```
ubuntu@ubuntu: /etc/apt
GNU nano 2.5.3          File: apt.conf
Acquire::http::Proxy "http://172.29.10.220:3128";

[ Read 2 lines (Warning: No write permission) ]
^G Get Help  ^O Write Out  ^W Where Is  ^K Cut Text  ^J Justify  ^C Cur Pos
^X Exit      ^R Read File  ^\ Replace   ^U Uncut Text  ^T To Spell  ^_ Go To Line
```

Ilustración 57. Añadiendo el proxy

- Instalando clang en nuestra máquina:

- Sudo apt-get install clang-3.6
- Exportamos el PATH:
 - En ipython:

```
clang.cindex.Config.set_library_file('/opt/1A/toolchain/x86_64-2.6.32-v2.0.17/lib/libc.so.6')
```
 - ```
clang.cindex.Config.set_library_file('/usr/lib/x86_64-linux-gnu/libclang-5.0.so.1')
```
- ```
python cindex-dump.py /remote/users/vsantiago/task5_vtmp/clang-example.py
```

Cada vez que se quiera trabajar con clang en nuestro proyecto se ha de añadir en el fichero las siguientes líneas para crear el índice que nos permita parsear los ficheros de código C++ y obtener las declaraciones de sus funciones en el fichero.

```
import clang
import clang.cindex
clang.cindex.Config.set_library_file('/opt/1A/toolchain/x86_64-2.6.32-v2.0.17/lib/libc.so.6')
idx = clang.cindex.Index.create()
```

Antes de seguir con la descripción del trabajo realizado en la maquina virtual la cual ha sido el entorno en el que se ha desarrollado este módulo, se va a describir también el trabajo que se ha realizado para este módulo en una máquina virtual en Docker donde los pasos seguidos han sido los siguientes:

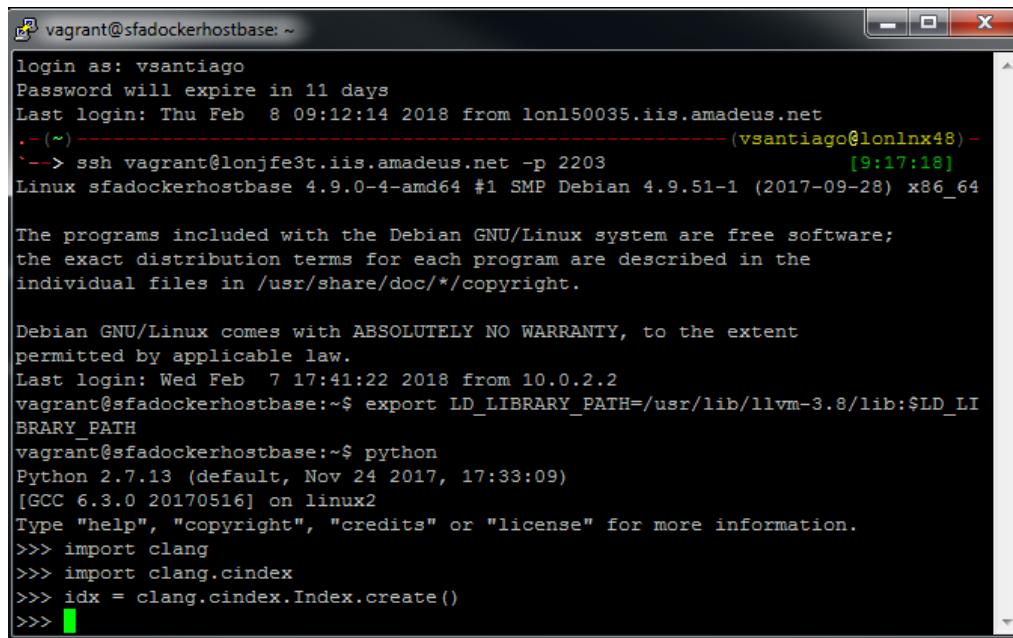
- **Máquina virtual en docker**

Ver la public key para añadir el acceso a la maquina virtual:

```
>cat ~/.ssh/id_rsa.pub
```

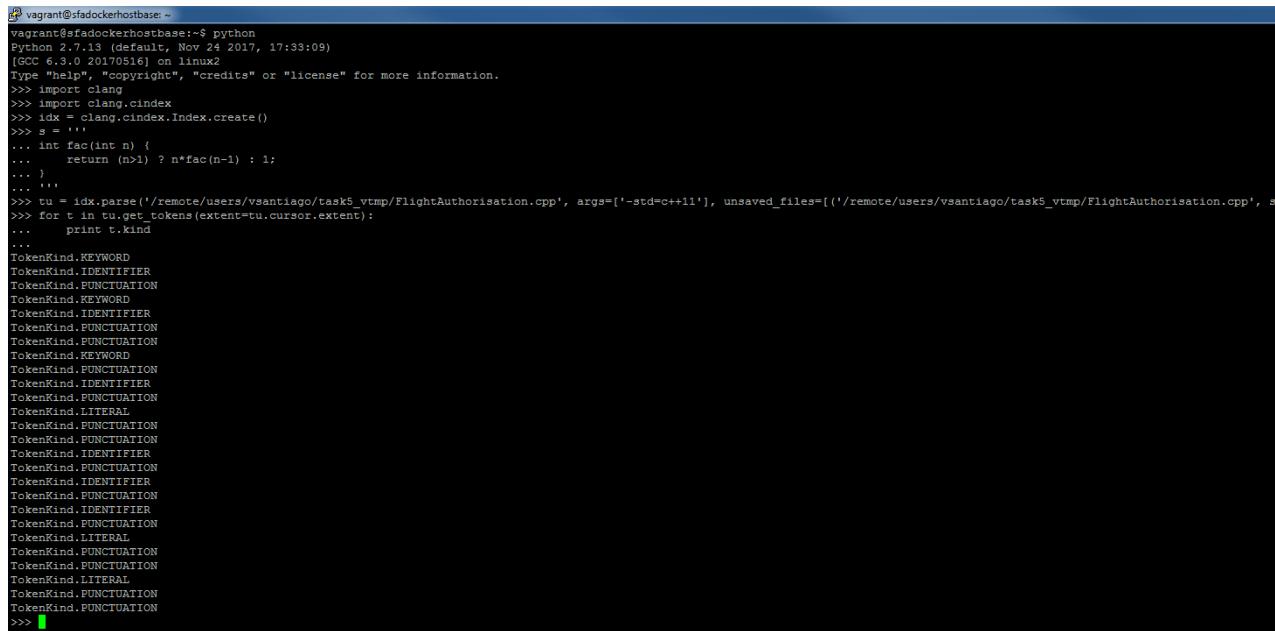
```
>ssh vagrant@lonjfe3t.iis.amadeus.net -p 2203  
>export LD_LIBRARY_PATH=/usr/lib/llvm-3.8/lib:$LD_LIBRARY_PATH  
>python
```

```
import clang  
import clang.cindex  
clang.cindex.Config.set_library_file('/opt/1A/toolchain/x86_64-2.6.32-v2 .  
0.17/lib/libc.so.6')  
idx = clang.cindex.Index.create()
```



```
vagrant@sfadockerhostbase: ~  
login as: vsantiago  
Password will expire in 11 days  
Last login: Thu Feb  8 09:12:14 2018 from lonl50035.iis.amadeus.net  
- - - (vsantiago@lonlnx48) - - -  
- - -> ssh vagrant@lonjfe3t.iis.amadeus.net -p 2203 [9:17:18]  
Linux sfadockerhostbase 4.9.0-4-amd64 #1 SMP Debian 4.9.51-1 (2017-09-28) x86_64  
  
The programs included with the Debian GNU/Linux system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/*/*copyright.  
  
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent  
permitted by applicable law.  
Last login: Wed Feb  7 17:41:22 2018 from 10.0.2.2  
vagrant@sfadockerhostbase:~$ export LD_LIBRARY_PATH=/usr/lib/llvm-3.8/lib:$LD_LIBRARY_PATH  
vagrant@sfadockerhostbase:~$ python  
Python 2.7.13 (default, Nov 24 2017, 17:33:09)  
[GCC 6.3.0 20170516] on linux2  
Type "help", "copyright", "credits" or "license" for more information.  
>>> import clang  
>>> import clang.cindex  
>>> idx = clang.cindex.Index.create()  
>>>
```

Ilustración 58. Máquina Docker funcionando.



```

vagrant@sfadockerhostbase: ~
vagrant@sfadockerhostbase:~$ python
Python 2.7.13 (default, Nov 24 2017, 17:33:09)
[GCC 6.3.0 20170516] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> import clang
>>> import clang.cindex
>>> idx = clang.cindex.Index.create()
>>> s = ''
... int fac(int n) {
...     return (n>1) ? n*fac(n-1) : 1;
... }
...
>>> tu = idx.parse('/remote/users/vsantiago/task5_vtmp/FlightAuthorisation.cpp', args=['-std=c++11'], unsaved_files=[('/remote/users/vsantiago/task5_vtmp/FlightAuthorisation.cpp', s)])
>>> for t in tu.get_tokens(extent=tu.cursor.extent):
...     print t.kind
...
TokenKind.KEYWORD
TokenKind.IDENTIFIER
TokenKind.PUNCTUATION
TokenKind.KEYWORD
TokenKind.IDENTIFIER
TokenKind.PUNCTUATION
TokenKind.PUNCTUATION
TokenKind.KEYWORD
TokenKind.PUNCTUATION
TokenKind.IDENTIFIER
TokenKind.PUNCTUATION
TokenKind.PUNCTUATION
TokenKind.LITERAL
TokenKind.PUNCTUATION
TokenKind.PUNCTUATION
TokenKind.IDENTIFIER
TokenKind.PUNCTUATION
TokenKind.IDENTIFIER
TokenKind.PUNCTUATION
TokenKind.IDENTIFIER
TokenKind.PUNCTUATION
TokenKind.IDENTIFIER
TokenKind.PUNCTUATION
TokenKind.LITERAL
TokenKind.PUNCTUATION
TokenKind.PUNCTUATION
TokenKind.LITERAL
TokenKind.PUNCTUATION
TokenKind.PUNCTUATION
TokenKind.PUNCTUATION
>>> █

```

Ilustración 59. Ejemplo de funcionamiento de Clang extrayendo los tokens.

En la imagen anterior se puede apreciar como se puede parsear por ejemplo el fichero FlightAuthorisation.cpp del cual por ejemplo se pueden apreciar los tokens anteriores.

Para ello hemos copiado el fichero local de nuestra máquina Ionlnx48 hacia la maquina por ssh usando:

```
scp vsantiago@ionlnx48:~/task5_vtmp/FlightAuthorisation.cpp .
```

El código ejecutado para obtener el nombre de las funciones por fichero cpp ha sido bastante parecido al siguiente:

```

import clang.cindex
from clang.cindex import *
clang.cindex.Config.set_library_file('/opt/1A/toolchain/x86_64-2.6.32-v2 .0.17/lib/libc.so.6')

function_name_list = []

def method_definitions(cursor):
    for i in cursor.walk_preorder():
        if i.kind != CursorKind.CXX_METHOD:
            if i.kind == CursorKind.FUNCTION_DECL:
                print i.kind, i.location, i.spelling
                function_name_list.append(i.spelling)
            continue
        if not i.is_definition():
            continue
        yield i
    return function_name_list

def extract_definition(cursor):
    filename = cursor.location.file.name
    with open(filename, 'r') as fh:
        contents = fh.read()
    return contents[cursor.extent.start.offset: cursor.extent.end.offset]

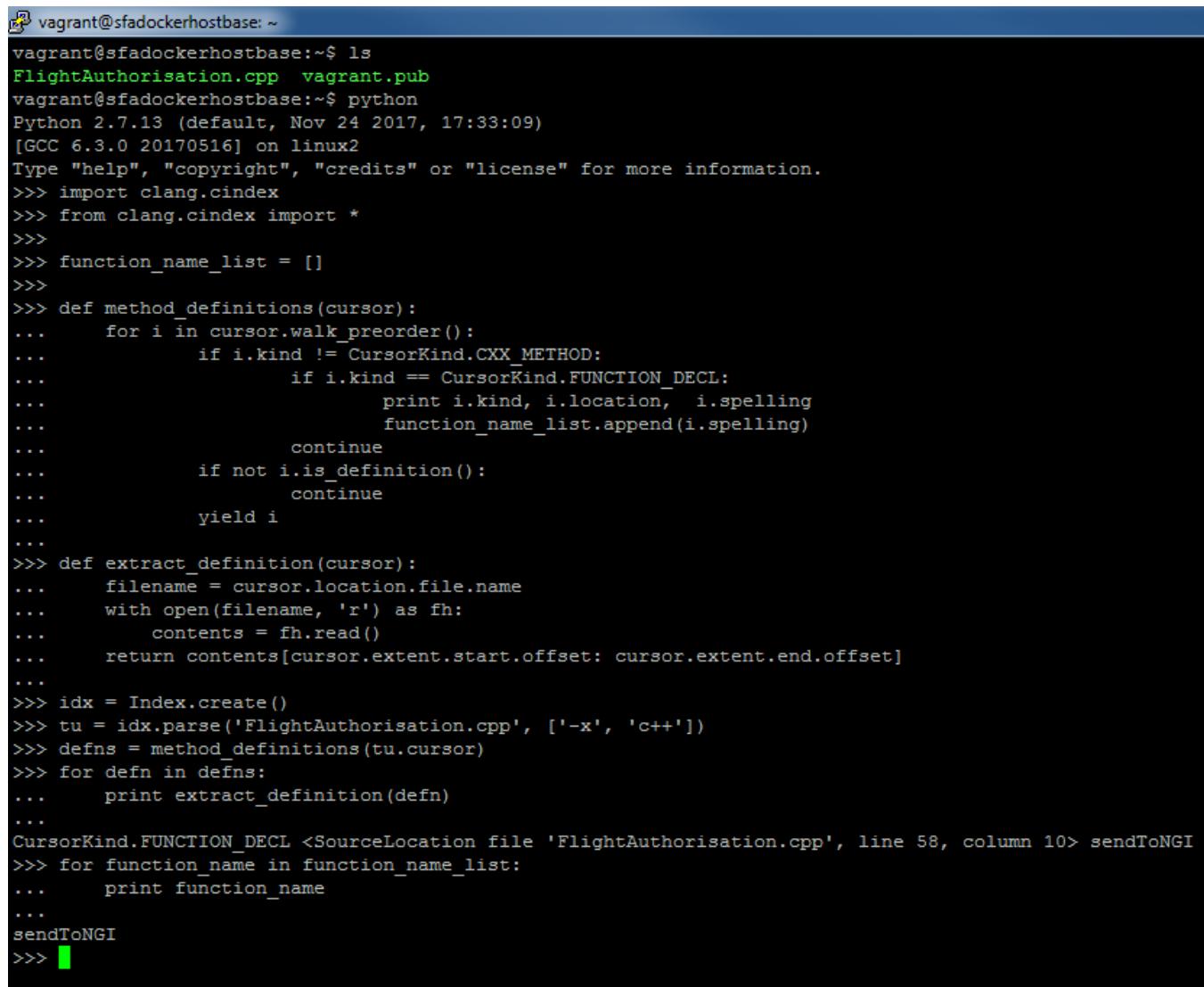
```

```

idx = Index.create()
tu = idx.parse('method.c', ['-x', 'c++'])
defns = method_definitions(tu.cursor)
for defn in defns:
    print extract_definition(defn)

```

El resultado es el nombre de la función (la función se llama sendToNGI) como se muestra en la siguiente captura de pantalla:



```

vagrant@sfadockerhostbase: ~
vagrant@sfadockerhostbase:~$ ls
FlightAuthorisation.cpp  vagrant.pub
vagrant@sfadockerhostbase:~$ python
Python 2.7.13 (default, Nov 24 2017, 17:33:09)
[GCC 6.3.0 20170516] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> import clang.cindex
>>> from clang.cindex import *
>>>
>>> function_name_list = []
>>>
>>> def method_definitions(cursor):
...     for i in cursor.walk_preorder():
...         if i.kind != CursorKind.CXX_METHOD:
...             if i.kind == CursorKind.FUNCTION_DECL:
...                 print i.kind, i.location, i.spelling
...                 function_name_list.append(i.spelling)
...                 continue
...             if not i.is_definition():
...                 continue
...             yield i
...
>>> def extract_definition(cursor):
...     filename = cursor.location.file.name
...     with open(filename, 'r') as fh:
...         contents = fh.read()
...     return contents[cursor.extent.start.offset: cursor.extent.end.offset]
...
>>> idx = Index.create()
>>> tu = idx.parse('FlightAuthorisation.cpp', ['-x', 'c++'])
>>> defns = method_definitions(tu.cursor)
>>> for defn in defns:
...     print extract_definition(defn)
...
CursorKind.FUNCTION_DECL <SourceLocation file 'FlightAuthorisation.cpp', line 58, column 10> sendToNGI
>>> for function_name in function_name_list:
...     print function_name
...
sendToNGI
>>> █

```

Ilustración 60. Obtención del nombre de la función sendToNGI del fichero FlightAuthorisaion.cpp

```

memory.txt      VoucherPrintStatus.cpp      db2.py      dbnodes.py      FlightAuthorisation.cpp      parse
43 #include <rtai-terminal.hpp>
44 #include <MsgHandler.hpp>
45
46 #include <FmtPublisherCreator.hpp>
47
48
49 using namespace FMT;
50 using namespace CPM;
51 using namespace CPS;
52 using namespace COM;
53 using namespace std;
54
55
56 namespace
57 {
58     bool sendToNGI(SviGen_FlightAuthorisation& iMsg, SviGen_OFlightAuthorisationResponse& oMsg)
59     {
60
61         SviOriginator& agent = iMsg.getOriginatorDetails();
62         Originator& org = COM::Originator::instance();
63
64         if( org.getCorporateId().isValid() ) { agent.setCorporateId( org.getCorporateId() ); }
65         if( org.getChannel().isValid() ) { agent.setChannel( org.getChannel() ); }
66         if( org.getCountry().isValid() ) { agent.setCountry( org.getCountry() ); }
67         if( org.getCurrencyPreference().isValid() ) { agent.setCurrencyPreference( org.getCurrencyPreference() ); }
68         if( org.getLanguagePreference().isValid() ) { agent.setLanguagePreference( org.getLanguagePreference() ); }
69         if( org.getAtid().isValid() ) { agent.setAtid( org.getAtid() ); }
70
71         if( org.getOfficeId().isValid() )
72         {
73             DCS_LOG_DEBUG("officeId:" << org.getOfficeId() );
74             agent.setOfficeId(org.getOfficeId());
75         }
76
77         if( org.getAuthorityCode().isValid() )
78         {
79             DCS_LOG_DEBUG("authorityCode:" << org.getAuthorityCode() );
80             agent.setAuthorityCode(org.getAuthorityCode());
81         }
82
83         DCS_LOG_TEST("NGI request : " << iMsg);
84
85         try
86         {

```

No results found for 'Index'

Ilustración 61. Contenido del fichero FlightAuthorisation.cpp

En la captura de pantalla anterior se puede aprecia el contenido del fichero cpp en el cual solo vemos declarada una función (sendToNGI) la cual hemos visto que ha sido parseada y devuelta con éxito usando Clang.

Uno de los problemas que se encontraron en este entorno consistía en que en ocasiones no devolvía el nombre de todas las funciones que se contenía el el fichero cpp por lo que finalmente se volvió a usar la máquina virtual normal que hemos descrito anteriormente ya que por falta de tiempo no se podía investigar más este detalle.

- **Máquina virtual Ubuntu 16:**

Imagen usada: Ubuntu 16.0.4.4

Como se ha mencionado anteriormente hay que establecer el proxy en el fichero apt.conf insertando la linea:

```
'Acquire::http::Proxy "http://yourproxyaddress:proxyport";'
```

A continuación se ha de exportar el proxy revisando primero la dirección IP de la maquina principal la cual en nuestro caso es 172.29.9.153.

También debemos de establecer en el fichero squid.conf el numero de puerto, en mi caso '*http_port 3126*' y reiniciar seguidamente Squid [20].

```
C:\ Select C:\windows\system32\cmd.exe
C:\Users\vsantiago>ipconfig

Windows IP Configuration

Ethernet adapter Bluetooth Network Connection:
  Media State . . . . . : Media disconnected
  Connection-specific DNS Suffix . . . . . : 

Ethernet adapter Local Area Connection:
  Connection-specific DNS Suffix . . . . . : lon.amadeus.net
  Link-local IPv6 Address . . . . . : fe80::a829:3fbd:caf0:115%11
  IPv4 Address . . . . . : 172.29.9.153
  Subnet Mask . . . . . : 255.255.252.0
  Default Gateway . . . . . : 172.29.8.1

Ethernet adapter VMware Network Adapter VMnet1:
  Connection-specific DNS Suffix . . . . . :
  Link-local IPv6 Address . . . . . : fe80::51cd:fd59:d00a:f7a4%17
  IPv4 Address . . . . . : 192.168.134.1
  Subnet Mask . . . . . : 255.255.255.0
  Default Gateway . . . . . : 

Ethernet adapter VMware Network Adapter VMnet8:
  Connection-specific DNS Suffix . . . . . :
  Link-local IPv6 Address . . . . . : fe80::1cc:7c00:89fe:6a8a%18
  IPv4 Address . . . . . : 192.168.19.1
  Subnet Mask . . . . . : 255.255.255.0
  Default Gateway . . . . . : 

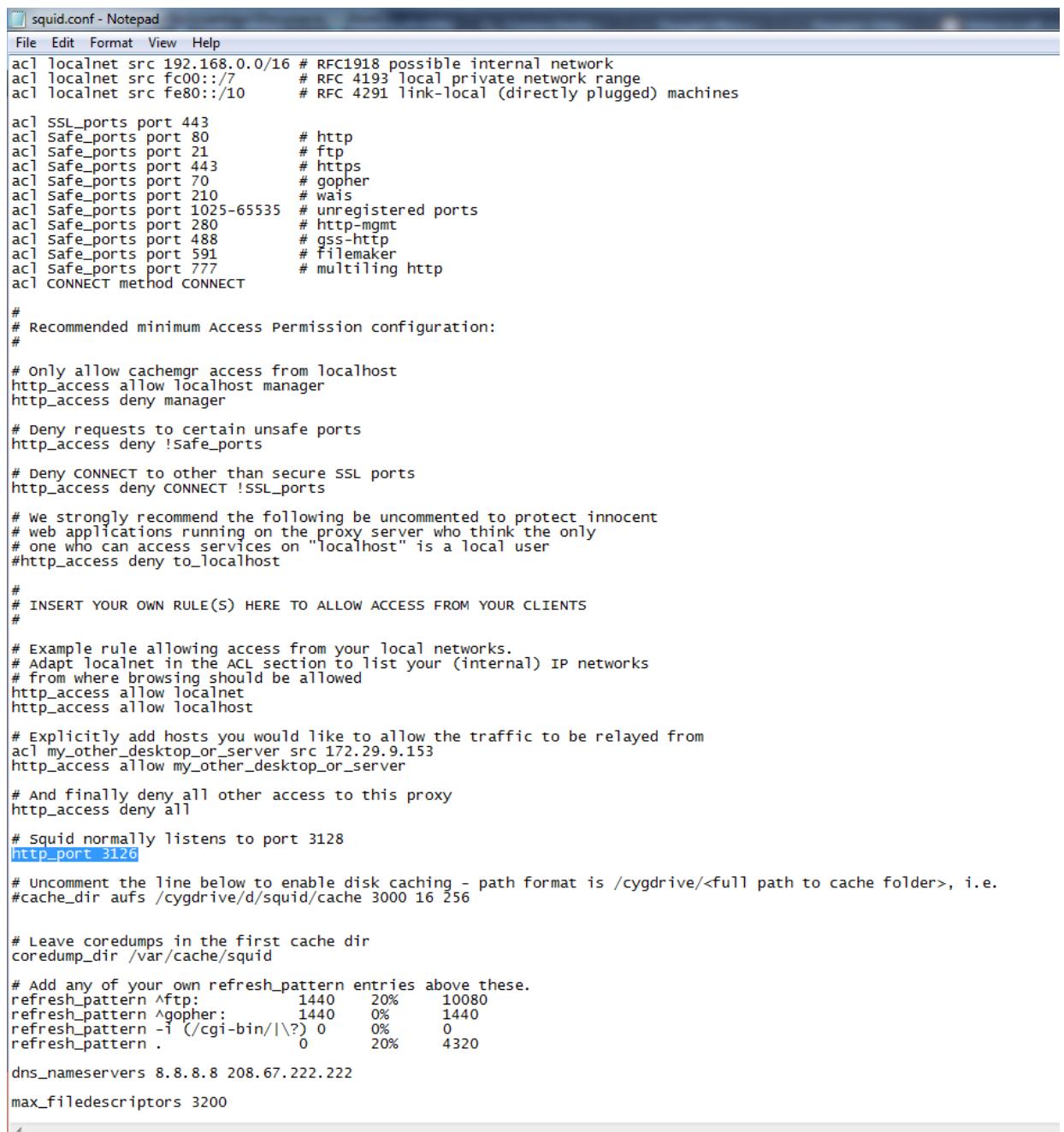
Tunnel adapter isatap.<7207601F-7604-4F11-9CA0-8C1CDD5A245E>:
  Media State . . . . . : Media disconnected
  Connection-specific DNS Suffix . . . . . : 

Tunnel adapter isatap.lon.amadeus.net:
  Media State . . . . . : Media disconnected
  Connection-specific DNS Suffix . . . . . : lon.amadeus.net

Tunnel adapter isatap.localdomain:
  Media State . . . . . : Media disconnected
  Connection-specific DNS Suffix . . . . . : 

C:\Users\vsantiago>
```

Ilustración 62. Salida de ipconfig



```

squid.conf - Notepad
File Edit Format View Help
acl localnet src 192.168.0.0/16 # RFC1918 possible internal network
acl localnet src fc00::/7      # RFC 4193 local private network range
acl localnet src fe80::/10     # RFC 4291 link-local (directly plugged) machines

acl ssl_ports port 443        # https
acl safe_ports port 80         # http
acl safe_ports port 21         # ftp
acl safe_ports port 443        # https
acl safe_ports port 70         # gopher
acl safe_ports port 210        # wais
acl safe_ports port 1025-65535 # unregistered ports
acl safe_ports port 280         # http-mgmt
acl safe_ports port 488        # gss-http
acl safe_ports port 591         # filemaker
acl safe_ports port 777        # multiling http
acl CONNECT method CONNECT

#
# Recommended minimum Access Permission configuration:
#
# Only allow cachemgr access from localhost
http_access allow localhost manager
http_access deny manager

# Deny requests to certain unsafe ports
http_access deny !safe_ports

# Deny CONNECT to other than secure SSL ports
http_access deny CONNECT !SSL_ports

# we strongly recommend the following be uncommented to protect innocent
# web applications running on the proxy server who think the only
# one who can access services on "localhost" is a local user
#http_access deny to_localhost

#
# INSERT YOUR OWN RULE(S) HERE TO ALLOW ACCESS FROM YOUR CLIENTS
#

# Example rule allowing access from your local networks.
# Adapt localnet in the ACL section to list your (internal) IP networks
# from where browsing should be allowed
http_access allow localnet
http_access allow localhost

# Explicitly add hosts you would like to allow the traffic to be relayed from
acl my_other_desktop_or_server src 172.29.9.153
http_access allow my_other_desktop_or_server

# And finally deny all other access to this proxy
http_access deny all

# Squid normally listens to port 3128
http_port 3128

# Uncomment the line below to enable disk caching - path format is /cygdrive/<full path to cache folder>, i.e.
#cache_dir aufs /cygdrive/d/squid/cache 3000 16 256

# Leave coredumps in the first cache dir
coredump_dir /var/cache/squid

# Add any of your own refresh_pattern entries above these.
refresh_pattern ^ftp:          1440   20%   10080
refresh_pattern ^gopher:        1440   0%    1440
refresh_pattern -i (/cgi-bin/|/\?) 0       0%    0
refresh_pattern .               0       20%   4320

dns_nameservers 8.8.8.8 208.67.222.222
max_filedescriptors 3200

```

Ilustración 63. Fichero de configuración de squid.

El siguiente paso es descargar los repositorios LLVM y Clang del repositorio de GitHub siguiendo las siguientes instrucciones [21]:

```

git clone https://github.com/llvm-mirror/llvm.git
cd llvm/tools
git clone https://github.com/llvm-mirror/clang.git

```

Seguidamente nos movemos a nuestro directorio y hacemos:

```
cd llvm  
.configure
```

o bien hacemos:

```
./configure --enable-target=x86_64,arm,cpp,cbe --with-built-clang --enable-pic --enable-polly --enable-shared --with-cxx-include-arch=x86_64 --enable-optimized --with-optimize-option=-Os --enable-assertions --disable-bootstrap --disable-multilib --enable-jit --enable-threads --no-create --no-recursion
```

Haciendo esto tendremos Clang en el siguiente path:

```
/home/Ubuntu/Desktop/llvm/tools/clang
```

A continuación, fuera del repositorio llvm creamos el directorio build en el cual se va a llevar la “construcción” de llvm y clang ejecutando cmake.

```
mkdir build  
cd build  
cmake -G "Unix Makefiles" .. llvm  
make
```

```
ubuntu@ubuntu:~/Desktop/build$ cmake -G "Unix Makefiles" .. llvm  
-- Could NOT find LibXml2 (missing: LIBXML2_LIBRARIES LIBXML2_INCLUDE_DIR)  
-- Native target architecture is X86  
-- Threads enabled.  
-- Doxygen disabled.  
-- Go bindings disabled.  
-- Could NOT find Ocaml (missing: OCAMLFIND OCAML_VERSION OCAML_STDLIB_PATH)  
-- Could NOT find Ocaml (missing: OCAMLFIND OCAML_VERSION OCAML_STDLIB_PATH)  
-- Ocaml bindings disabled.  
-- Could NOT find Python module pygments  
-- Could NOT find Python module pygments.lexers.c_cpp  
-- Found Python module yaml  
-- LLVM host triple: x86_64-unknown-linux-gnu  
-- LLVM default target triple: x86_64-unknown-linux-gnu  
-- Building with -fPIC  
-- Constructing LLVMBuild project information  
-- Linker detection: GNU ld  
-- Targeting AArch64  
-- Targeting AMDGPU  
-- Targeting ARM  
-- Targeting BPF  
-- Targeting Hexagon  
-- Targeting Lanai  
-- Targeting Mips  
-- Targeting MSP430  
-- Targeting NVPTX  
-- Targeting PowerPC  
-- Targeting Sparc  
-- Targeting SystemZ  
-- Targeting X86  
-- Targeting XCore  
-- Clang version: 7.0.0  
-- Configuring done  
-- Generating done  
-- Build files have been written to: /home/ubuntu/Desktop/build  
ubuntu@ubuntu:~/Desktop/build$ ls  
bin CMakeFiles CPackConfig.cmake DummyConfigureOutput lib Makefile test utils  
cmake cmake_install.cmake CPackSourceConfig.cmake examples projects tools  
CMakeCache.txt compile_commands.json docs include llvm.spec runtimes unitests
```

Esto contruye ambos, tanto LLVM como Clang en su modo debug.

Nota: para el posterior desarrollo de Clang, puede ejecutar make clang.

Se podrán consultar los ejecutables llvm & clang después de terminar todo el proceso de build.

El proceso normalmente toma cerca los los 10 - 20 minutos.

Durante este paso se encontraron varios problemas como es el siguiente:

```
/usr/bin/ld: error: ../../lib/libLLVMSelectionDAG.a: ELF section name out of range collect2: error: ld returned 1 exit status  
tools/lto/CMakeFiles/LTO.dir/build.make:269: recipe for target 'lib/libLTO.so.7.0.0svn' failed make[2]: * [lib/libLTO.so.7.0.0svn] Error 1  
CMakeFiles/Makefile2:18957: recipe for target  
'tools/lto/CMakeFiles/LTO.dir/all' failed make[1]: * [tools/lto/CMake-  
Files/LTO.dir/all] Error 2 Makefile:149: recipe for target 'all' failed make:  
*** [all] Error 2
```

Tras investigar el problema se llegó a la conclusión de que se debía a que el tamaño de la memoria de la máquina no era suficiente. La más rápida y eficaz solución fue incrementar el tamaño de la memoria para /dev/sda1 en nuestra máquina.

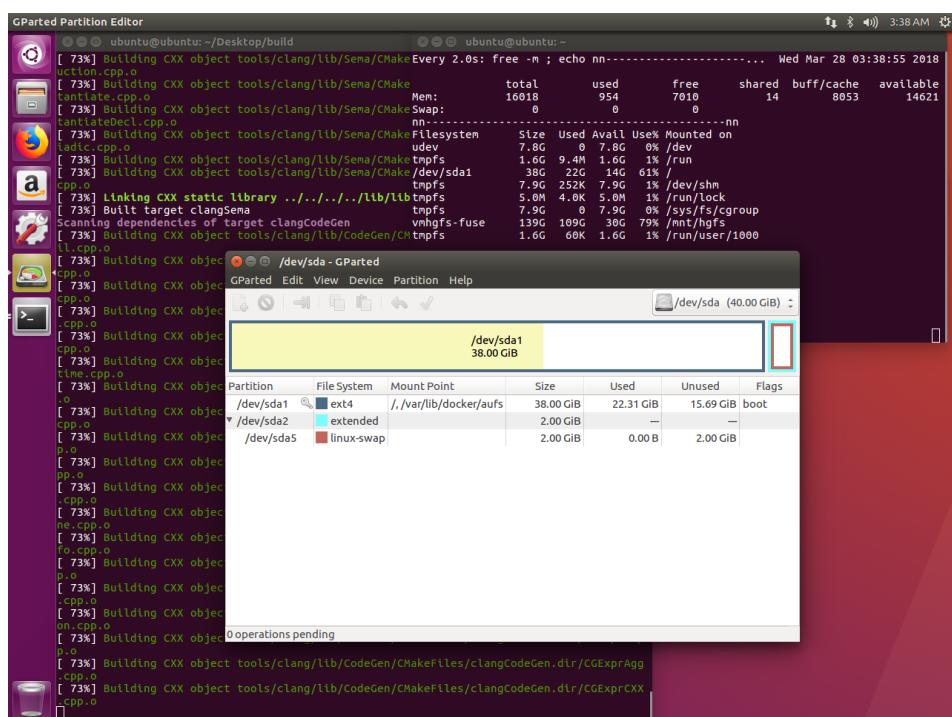


Ilustración 65. Resize de la memoria

Tras solventar este problema lanzamos la compilación de nuestro target donde podemos ver que se encuentra en el 92% por lo que a continuación únicamente queda ejecutar la librería pasándole el fichero a pararse con el siguiente comando:

```
clang -Xclang -load -Xclang lib/PrintFunctionNames.so hello-world.cpp
```

Seguidamente, si no hemos exportado a PATH nuestros repositorios, obtendremos el siguiente error al intentar cargar el plugin:

Problemas encontrado al cargar el plugin:

```
ubuntu@ubuntu:~/Desktop/build$ clang++ -D_GNU_SOURCE -DDE-BUG -D_STDC_CONSTANT_MACROS -D_STDC_FORMAT_MACROS -D_STDC_LIMIT_MACROS -D_GNU_SOURCE -I$BD/tools/clang/include -Itools/clang/include -I$BD/include -linclude .. llvm/tools/clang/tools/clang-check/Clang-Check.cpp -fsyntax-only -Xclang -load -Xclang $BD/lib/PrintFunctionNames.so -Xclang -plugin -Xclang print-fns
```

error: unable to load plugin '/home/ubuntu/Desktop/build/lib/PrintFunctionNames.so': '/home/ubuntu/Desktop/build/lib/PrintFunctionNames.so': undefined symbol: _ZN4llvm23EnableABIBreakingChecksE'

```
ubuntu@ubuntu:~/Desktop/build$ clang -Xclang -load -Xclang lib/PrintFunctionNames.so hello-world.cpp
```

error: unable to load plugin 'lib/PrintFunctionNames.so': 'lib/PrintFunctionNames.so': undefined symbol: _ZN4llvm23EnableABIBreakingChecksE'

Tras completar el built de los targets e intentar ejecutar el fichero a pararse sin éxito, procedemos a exportar en PATH la localización de nuestro directorio LLVM y Clang de la siguiente forma para solventar el error obtenido previamente [23]:

```
ubuntu@ubuntu:~/Desktop$ export  
LLVM_DIR=/home/ubuntu/Desktop/llvm
```

```
ubuntu@ubuntu:~/Desktop$ export  
CLANG_DIR=/home/ubuntu/Desktop/llvm/tools/clang
```

```
ubuntu@ubuntu:~/Desktop$ export PATH=$PATH:$  
{LLVM_DIR}/install/bin
```

```
ubuntu@ubuntu:~/Desktop$ export PATH=$PATH:$  
{CLANG_DIR}/install/bin
```

Y ahora hacemos de nuevo el cmake:

```
cmake -Drpath=ON \
-DExplicitlink=ON \
-Dxrootd=OFF \
-D_builtin-freetype=OFF \
-DCMAKE_CXX_COMPILER=/usr/local/products/gcc/5.2.0-nogold/bin/g++
+ \
-DCMAKE_C_COMPILER=/usr/local/products/gcc/5.2.0-nogold/bin/gcc \
-DCMAKE_INSTALL_PREFIX=my_directory \
-DCMAKE_CXX_FLAGS=-D_GLIBCXX_USE_CXX11_ABI=0 .../llvm
```

Seguidamente procedemos a hacer el build con make construyendo solamente nuestro target:

```
make PrintFunctionNames
```

A continuación solo tendremos que ejecutar de nuevo la clang con el fichero a parsear [24].

Tras tener bien configurada y corriendo la librería en nuestra máquina, se procedió a crear finalmente el fichero .py el cual tiene como fin crear un fichero JSON en el que se almacenan por cada fichero cpp de código nuevo o modificado, recolectar las funciones que hay en su interior las cuales serán las mismas que nos van a decir que sets de regresión del módulo 1 se han de ejecutar para testear estos ficheros cpp modificados.

El directorio build querdaría como se muestra en la siguiente imagen:

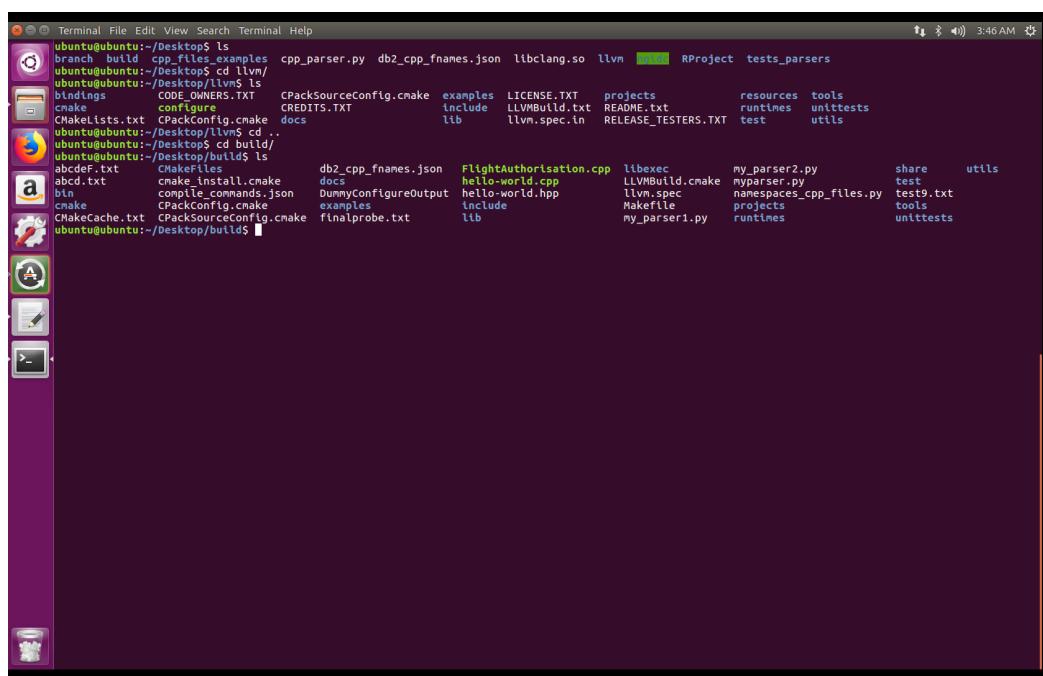


Ilustración 66. Directorio build

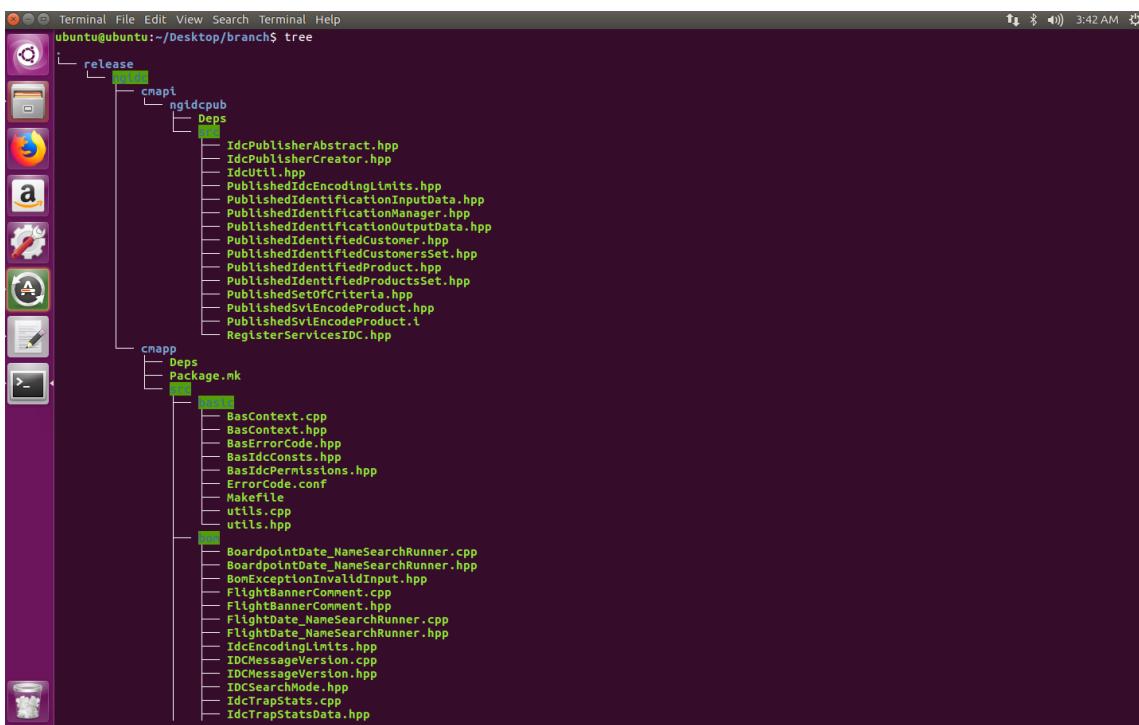


Ilustración 67. Árbol con ficheros cpp y sus_hpp

En la imagen anterior se muestra como por ejemplo en el directorio cmapp se encuentran los ficheros.

```
import print
import clang
from clang.cindex import *
import clang.cindex as cl
import sys
import json
import os
from os import walk

original_path = '/home/ubuntu/Desktop/branch/release/nglcpub/cmapp'

def print_indent(level):
    for i in range(0,level):
        sys.stdout.write(' ')
    sys.stdout.flush()

def get_functions(node, level, funcs, store): # start with -1, False
    if node.kind == clang.cindex.CursorKind.NAMESPACE \
            and (node.displayname == '__gnu_cxx' or node.displayname == '__gnu_debug') \
            or node.kind == clang.cindex.CursorKind.UNEXPOSED_DECL or ('__' in node.displayname):
        return

    if node.kind == clang.cindex.CursorKind.NAMESPACE:
        print_indent(level)
        store = True
        funcs.append([])
        print_level, node.kind, node.displayname

    elif node.kind == clang.cindex.CursorKind.FUNCTION_DECL or node.kind == clang.cindex.CursorKind.CXX_METHOD:
        print_indent(level)
        #print_level, node.kind, node.displayname
        if store:
            funcs[len(funcs)-1].append(node.displayname)

    for child in node.get_children():
        get_functions(child, level+1, funcs, store)

def get_cpp_namespaces(PATH_CPP_FUNC):
    namespaces_list = []
    with open(PATH_CPP_FUNC, 'r') as file:
        functions = file.read()
        file_by_lines = functions.split('\n')

    for line in file_by_lines:
        if "namespace" in line and "using" not in line and "://" not in line:
            namespaces_list.append(line)

    return namespaces_list
```

Ilustración 68. Ejemplo del código del fichero parseador.

```
Z:\tasks\zmp\prj\cpp\project\eng_persontob_Cpp\FNames.json - Notepad++  
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?  
regressions.dbx db1_Reg_FNames.json db2_Cpp_FNames.json db3_regressions_to_execute_old.json db4_aaaaibccp.btx db3.regressions_to_execute.json db2.py db1_Reg_FNamesOld.json  
1 [{}]  
2 ["BMT": {"CMD_Divide": "getFrequentTravellerData()", "buildIdentifiedProducts(int 6)", "clearIdentifiedProducts(int 6)", "identifyCustomerAndProduct(const COM::PublishedIdentificationInputData &)", "COM::PublishedIdentificationInputData &"}, "IDC": {"CMD_Divide", "obje::NGDCallbackService::~NGDCallbackService()", "setIssuingCarrierNumber(COM::FldBticCarrierNumber, const&)", "DCS_DEF_C_SRC()"}, "BMT": {"setIssuingCarrierNumber(COM::FldBticCarrierNumber, const&)"}, "SMT": {"getFrequentTravellerData()", "ip_local_deliver_"}]  
3  
4  
5  
6
```

Ilustración 69. Ejemplo donde se obtienen en el JSON los nombres de las funciones por fichero cpp

A modo de resumen, en este módulo lo que se ha desarrollado ha sido la parte de la herramienta que se encarga de obtener, desde una lista de ficheros cpp a los cuales se les ha añadido o modificado algo, las funciones que contienen con el fin de almacenar estos nombres de función y sus dependencias en el fichero JSON.

En el fichero JSON tendremos definidas como claves el path hacia el fichero cpp y su nombre y como valores la lista de funciones contenidas en el como muestra la siguiente imagen donde se pueden ver tres ficheros cpp y sus correspondientes listas de funciones agrupadas a su vez por namespaces.

Ilustración 70. Fichero JSON del módulo 2

5.4. MÓDULO 3: Subsets de regresiones implicados.

Este módulo es el encargado de combinar e integrar los otros módulos para obtener como resultado una lista de subsets de regresiones a ejecutar para testear el código que el usuario ha modificado.

El procedimiento sería a groso modo como el que se muestra en el siguiente diagrama:

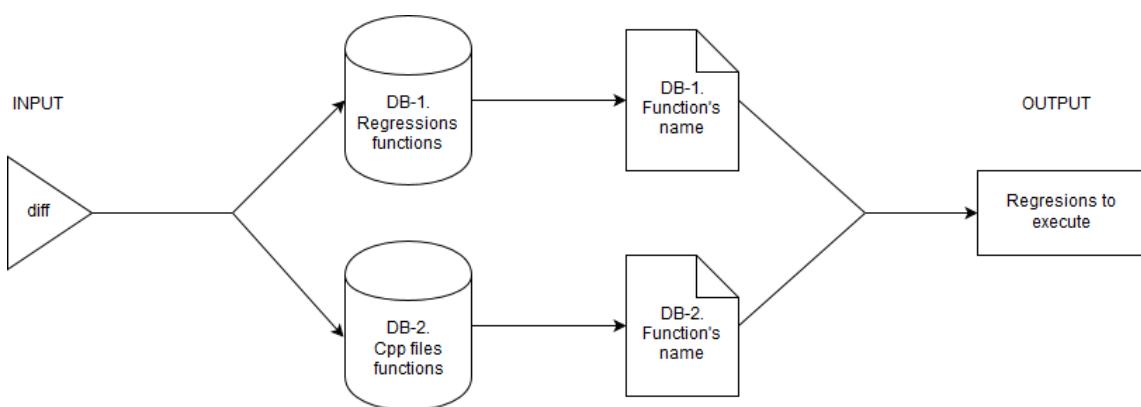


Ilustración 71. Procedimiento del módulo 3.

Se basa fundamentalmente en general los ficheros json tanto del módulo 1 como del módulo 2 y realizar el mapeo de ambos.

Básicamente lo que hace es buscar las funciones que se tienen en el fichero json del modulo 2 el cual contiene únicamente las funciones del código c++ que ha sido modificada y realizar su búsqueda en el fichero json del módulo 1 el cual tenía todas las funciones las cuales eran testeadas por los distintos subsets de regresiones.

Con ello lo que se obtiene es que si solo se han modificado por ejemplo 3 funciones de un fichero cpp del proyecto, solo se ejecuten los pequeños subset de regresión que afectan a esta parte del código por lo que con ello se ahorra mucho tiempo ya que no se ejecuta la suite entera de regresiones sino que simplemente se ejecutan las que necesitamos.

Con ello nos permite realizar las pruebas rápidamente y poder con ello obtener los resultados en poco tiempo.

A continuación la siguiente ilustración muestra parte del código usado para ello.

The screenshot shows the Atom code editor with the file `db3_reg_to_execute.py` open. The code is a Python script designed to compare two JSON files (db1 and db2) and identify regression functions. It uses the `json` module to load the files, iterates through each namespace in db2, and checks if any function in db2 has been removed from db1. If so, it adds the subset to a list of regressions. Finally, it generates a JSON file containing these regressions.

```

db3_reg_to_execute.py — Z:\task5_vtmp\db3-MapReg — Atom
File Edit View Selection Find Packages Help
Project
Documents
Custom Office Templates
Dell Downloads
My Music
My Pictures
My Videos
MyLabberChats
MyLabberFiles
OneNote Notebooks
VictoriaDocs
Virtual Machines
desktop.ini
memory.txt
memory.txt
1 import csv
2 import json
3
4 def get_regressions_to_execute(db1_path, db2_path):
5     regressions_to_execute = []
6     db1_file = json.loads(open(db1_path).read())
7     db2_file = json.loads(open(db2_path).read())
8     for each_namespace, each_function in db2_file.items():
9         for function_name2 in db2_file[each_namespace]:
10            for subsets in db1_file.keys():
11                for each_value in db1_file[subsets]:
12                    function_name1 = each_value[-1]
13                    if function_name1.find(function_name2)>=0 and subsets not in regressions_to_execute:
14                        regressions_to_execute.append(subsets)
15                        print function_name1, "----", function_name2, "-----", each_namespace, "-----", subsets
16
17    return regressions_to_execute
18
19 def generate_json_file(regressions_list):
20     dictionary = {'regressions': regressions_list}
21     dictionary_to_json = json.dumps(dictionary)
22     with open('db3_regressions_to_execute.json', 'w') as outfile: #set a path where it generates the file
23         outfile.write(dictionary_to_json)
24
25
26 def main():
27     db1_path = '/remote/users/vsantiago/task5_vtmp/db1-RegFunc/db1_Reg_FNames.json'
28     db2_path = '/remote/users/vsantiago/task5_vtmp/db2-CppFunc/clang_parser/db2_Cpp_FNames.json'
29
30     regressions = get_regressions_to_execute(db1_path, db2_path)
31     generate_json_file(regressions)
32     print "\nRegressions to execute: ", regressions
33
34 if __name__ == "__main__":
35     main()

```

Ilustración 72. Captura del código correspondiente al módulo 3.

Finalmente tras ejecutar este modulo, se devuelve al usuario de regresión únicamente los subsets que ha de ejecutar como muestra la siguiente captura de pantalla en la cual también se ha incluido para su mejor entendimiento las funciones que han sido buscadas el el fichero JSON del módulo 2 con sus correspondientes funciones encontradas en el fichero JSON del módulo 1 del proyecto.

The terminal session shows the execution of the `db3_reg_to_execute.py` script. It first lists the contents of the current directory, then runs the script with the arguments `db3_regressions_to_execute.json` and `db3_regressions_to_execute_old.json`. The output shows the regression functions found between the two versions. The session ends with a prompt for the user to press Ctrl+C to exit.

```

.~/task5_vtmp/db3-MapReg
.~> ls
db3_reg_to_execute.py  db3_regressions_to_execute.json  db3_regressions_to_execute_old.json
.~/task5_vtmp/db3-MapReg
.~> python db3_reg_to_execute.py -- /remote/users/vsantiago/task5_vtmp/callgraph_to_csv/opreport2neo4j/func_csv_files/func_oprofile.out.RGR_LVP.1524649069_call_graph.csv
.~> CMP_Divide ---- CMP_Divide ----- /remote/users/vsantiago/task5_vtmp/callgraph_to_csv/opreport2neo4j/func_csv_files/func_oprofile.out.RGR_LVP.1524649069_call_graph.csv
.~> ip_local_deliver_start ---- ip_local_deliver ----- SMT ----- /remote/users/vsantiago/task5_vtmp/callgraph_to_csv/opreport2neo4j/func_csv_files/func_oprofile.out.RGR_IDC.1524648912_call_graph.csv
ip_local_deliver_finish ---- ip_local_deliver ----- SMT ----- /remote/users/vsantiago/task5_vtmp/callgraph_to_csv/opreport2neo4j/func_csv_files/func_oprofile.out.RGR_IDC.1524648912_call_graph.csv
Regressions to execute: ['u'/remote/users/vsantiago/task5_vtmp/callgraph_to_csv/opreport2neo4j/func_csv_files/func_oprofile.out.RGR_LVP.1524649069_call_graph.csv', u'/remote/users/vsantiago/task5_vtmp/callgraph_to_csv/opreport2neo4j/func_csv_files/func_oprofile.out.RGR_IDC.1524648912_call_graph.csv']
.~/task5_vtmp/db3-MapReg
.~>
[0] 0:zsh lizah* 3:zsh-

```

Ilustración 73. Salida de la ejecución del Módulo 3.

A modo de resumen, el proceso seguido desde que se inserta el “diff” con el código modificado hasta que se obtienen los subsets de regresiones a ejecutar es el que se muestra en esta ilustración:

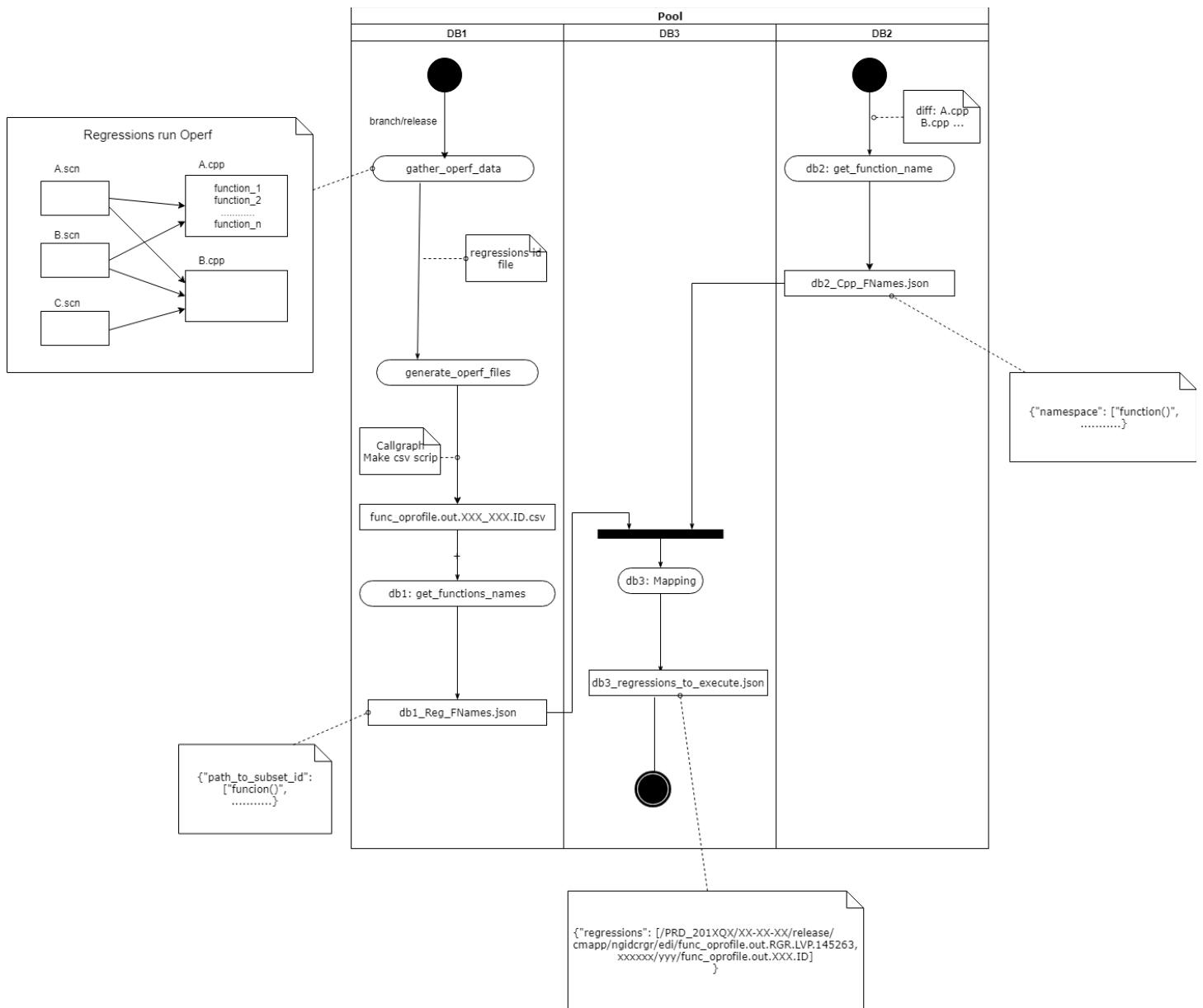


Ilustración 74. Procedimiento realizado al ejecutar la herramienta para obtener las regresiones a ejecutar

6. Manual de usuario

A continuación en este apartado se va a describir brevemente el uso de las herramientas desarrolladas comenzando por ejemplo con la aplicación web.

6.1. Dashboard de nuevos usuarios de regresión

Esta sección documenta el proceso que se ha de seguir para usar la dashboard de usuarios de regresión siguiendo los siguientes puntos:

- Breve descripción
- Introducción a su uso
- Lanzando la dashboards
- Usando el menú principales
- Lista de opciones de usuario

6.1.1. Breve descripción

El objetivo principal de este panel de control es crear fácilmente nuevos usuarios agregando su correspondiente información sobre Unix Id, Machine, nombre de usuario de la base de datos, base de datos y puerto.

Además, la dashboard ofrece la posibilidad de listar, actualizar, eliminar y filtrar usuarios.

6.1.2. Introducción a su uso

Se han de seguir los siguientes pasos:

1. Para acceder al tablero desde el navegador, solo se ha de introducir la siguiente url en el navegador:

<http://ngdwiki.lon.amadeus.net/dashboard/>

2. Seguidamente se ha de hacer click en la opción Panel de control de regresión y el panel de usuario de Regresión estará listo para su uso.

6.1.3. Lanzar el Tablero

Usando el menú principal se han de elegir una de las siguientes opciones del menú principal:

- Crear usuarios de regresión.
- Listar usuarios.
- Importar desde el repositorio de usuarios
- Sincronizar repositorio
- Sincronizar cambios
- Exportar a CVS

A continuación en los siguientes apartados se explica brevemente las distintas acciones que se pueden realizar sobre los usuarios.

1. Crear usuarios de regresión.

Usando esta vista podemos crear un nuevo usuario especificando como campos obligatorios su "Unix ID" y la "Máquina" en la que trabaja. Seguidamente se ha de hacer click en el botón "Enviar" para poder crear el nuevo usuario.

Si hacemos click en "Opciones avanzadas", nos mostrará el menú desplegable con las siguientes opciones para agregar:

- Nombre de usuario DB
- Base de datos
- Puerto

En la siguiente imagen se muestra la aplicación ya desplegada en la maquina usada por el equipo de Software Framework Architecture el cual como se ha mencionado anteriormente, es el encargado de realizar estas tareas.

En la imagen se aprecia la vista principal la cual nos lleva a crear estos nuevos usuarios.

← → C ⓘ ionlnx48:40000/new_comer/create_user/#

Regression users dashboard | Create

Log in - Help -

Create regressions users

Unix ID:

Machine:

Advanced options -

DB Username:

Database: ▾

Port:

[List users](#) [Import form users repository](#) [Synchronize changes](#) [Export to CSV](#)

Regression users dashboard © 2017

Iustración 75. Ventana de creación de nuevos usuarios de regresión.

2. Listar usuarios de regresión.

Este botón te permite mostrar la lista de todos los usuarios que se describe en el siguiente punto. (Lista de la opción del usuario).

3. Importar del repositorio de usuarios

Al hacer click en este botón se podrán llevar a cabo las acciones de actualizar a los usuarios en nuestra base de datos agregando los nuevos usuarios del archivo "users_x" almacenado en nuestro repositorio de Mercurial (int / ngtoolrc / users / users_x).

4. Sincronizar el repositorio

El objetivo principal de esta opción es generar / sincronizar el archivo "user.-conf" que se tiene en mercurial con la información que tenemos añadida al igual que llevar a cabo las tareas de sincronización en las máquinas.

5. Exportar a CVS

Con esta opción, se puede llevar a cabo la exportación de la lista de datos de los usuarios a un archivo csv delimitado por comas, que luego se puede usar para importar a una hoja de cálculo o base de datos.

A continuación detallada ya la primera vista general, pasamos a detallar la opción de listar usuarios ya existentes donde podremos hacer el filtrado, las modificaciones necesarias a cada usuario, también se podrán hacer aquí las migraciones y la eliminación de usuarios.

Regression users dashboard | Comers's List

Latest regression users

ID	Unix ID	DB Username Peak 1	DB Username Peak 2	Database	Port	Update	Delete
4014	mraimold	mraimold	mraimold_p2	LONDB04	100		
4021	skopcsa	skopcsa	skopcsa_p2	LONDB04	18		
4020	mvaikus	mvaikus	mvaikus_p2	LONDB04	17		
4019	mgocieva	mgocieva	mgocieva_p2	LONDB04	15		
4018	qbernard	qbernard	qbernard_p2	LONDB04	111		
4017	jkong	jkong	jkong_p2	LONDB04	136		
4016	zhamilton	zhamilton	zhamilton_p2	LONDB04	60		
4015	rmillward	rmillward	rmillward_p2	NCMUSR04LS	66		
4022	lhopkins	lukehopkins	lukehopkins_p2	LONDB04	22		
4013	vsantiago	vsantiago	vsantiago_p2	LONDB04	174		
3933	rdagenais	rdagenais	rdagenais_p2	LONDB04	203		
3932	rblusr2	rblusr2	rblusr2_p2	LONDB04	177		
3931	rblusr1	rblusr1	rblusr1_p2	LONDB04	177		
3930	cdevautour	cdevautour	cdevautour_p2	LONDB04	176		
3929	jacopo.biscella	jacopo.biscella	jacopo.biscella_p2	NCMUSR04LS	153		
3928	daniel.hurtado	daniel.hurtado	daniel.hurtado_p2	LONDB04	240		
3927	pannamalai	pannamalai	pannamalai_p2	NCMUSR02LS	227		
3926	anand.palani	anand.palani	anand.palani_p2	LONDB04	213		
3925	psathineni	psathineni	psathineni_p2	LONDB04	212		

Options:

- Select / Deselect All
- Migrate to: LONDB04
- General filter:

[New regression user](#) [Import from users repository](#) [Synchronize changes](#) [Export to CSV](#)

Ilustración 76. Vista de la lista de usuarios existentes

■ Herramientas de la parte superior derecha:

- Seleccionar / Deseleccionar todos y Migrar

Options:

Select / Deselect All

Migrate to: LONDB04

General filter:

LONDB04
 LOCALDB
 NCMUSR02LS
 NCMUSR03LS
 NCMUSR04LS
 NCMUSR05LS

Ilustración 77. Opciones de migración.

- Filtro general.

Donde se puede apreciar en la siguiente imagen que al introducir por ejemplo el número 33, realiza la búsqueda en todos los campos y arroja únicamente los usuarios de regresión que contienen ese número en alguno de sus campos ya sea en su ID, puerto o incluso Unix ID o nombre en las bases de datos.

ID	Unix ID	DB Username Peak 1	DB Username Peak 2	Database	Port	Update	Delete
3933	rdagenais	rdagenais	rdagenais_p2	LONDB04	203		
3934	oleksii.novikov	oleksiinovikov	oleksiinovikov_p2	LONDB04	33		
3895	hsenussi	hsenussi	hsenussi_p2	LONDB04	33		
3833	jjose	jjose	jjose_p2	LONDB04	63		
3633	dimitrios.vikeloudas	dimitriosvikeloudas	dimitriosvikeloudas_p2	LONDB04	43		
3838	pawel.rozenberg	pawelrozenberg	pawelrozenberg_p2	NCMUSR02LS	133		

Options:

- Select / Deselect All
- Migrate to: LONDB04
- General filter:

Ilustración 78. Ejemplo de filtro general

- Filtro por campos.

Como se puede apreciar en la siguiente captura de pantalla, se han filtrado los usuarios de regresión únicamente por el campo Unix ID en el cual como se ve en el ejemplo se han ingresado las letras "vs" por lo que se nos ha devuelto únicamente los usuarios que contienen dichas letras en su Unix ID.

ID	Unix ID	DB Username Peak 1	DB Username Peak 2	Database	Port	Update	Delete
	vs						
4013	vsantiago	vsantiago	vsantiago_p2	LONDB04	174		
3997	vsoomaneyvijaykumar	vsoomaneyvijaykumar	vsoomaneyvijaykumar_p2	LONDB04	68		

Ilustración 79. Ejemplo de filtro por campos.

- Modificar / Actualizar datos del usuario de regresión.

A continuación en esta ilustración se nos muestra como es la vista de actualizado de contenidos en la cual se da la posibilidad de teniendo un usuario, poder actualizar los campos del mismo únicamente ingresándolos de nuevo en el formulario y pulsando en el botón “Update” el cual envía la nueva información registrada y actualiza nuestro usuario en el sistema.

Regression users dashboard | Update

Update user

ID: 4013

Unix ID:

DB Username Peak 1:

DB Username Peak 2:

Database: LONDB04 (current selection)

Port:

[Create comers](#) [List comers](#) [Delete](#)

Ilustración 80. Ejemplo de Actualización/Modificación de datos de un usuario.

- Eliminación de un usuario de regresión.

El siguiente procedimiento solo consiste en seleccionar un usuario de la lista de usuarios existentes y por medio del botón “Delete” realizar su eliminado desde la siguiente ventana la cual requiere la confirmación del usuario administrador de nuevo para completar el procedimiento de borrado de dicho usuario seleccionado mostrándole sus detalles.

A continuación en la siguiente imagen se tiene un ejemplo donde se ve se está borrando el usuario “vsantiago”.

Regression users dashboard | Delete

Delete user

ID: 4013
Unix ID: **vsantiago**
DB Username Peak 1: **vsantiago**
DB Username Peak 2: **vsantiago_p2**
Database: **LONDB04**
Port: **174**

[Delete](#)

[Create comers](#)

[List comers](#)

Ilustración 81. Ejemplo de Eliminar usuario de regresión.

6.2. End-to-End Functional Test coverage Tool

Detallada ya la web de ingreso de usuarios de regresión los cuales van a ejecutar los test de regresión, se procede a describir en este subapartado el funcionamiento de la herramienta que se ha desarrollado la cual permite ejecutar únicamente el subset de regresiones que afecta directamente a los cambios realizados en el software del producto principal de la compañía, reduciendo con ello sus costes, tiempo de ejecución, eficiencia y obtención temprana de resultados.

Se ha realizado un video en el cual se muestra su perfecto funcionamiento el cual se encuentra en el siguiente enlace (también adjuntado en la documentación).

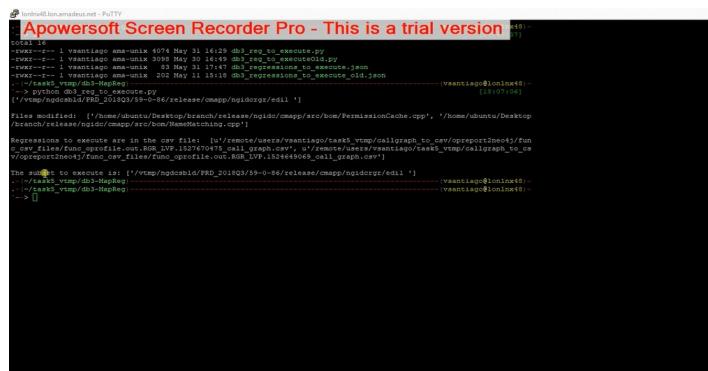


Ilustración 82. Video.

En cuanto a los resultados obtenidos, decir que se han cumplido todos los objetivos fijados, y a modo de evaluación cuantitativa, comentar que se ha reducido significativamente el coste de tiempo con ambas herramientas en la primera reduciendo el tiempo hasta al 95% puesto que con una acción de click en un botón simplemente realiza todas las tareas de inserción y sincronización, también se ve reducida al 95% el tiempo de modificación y eliminado.

En el caso de la herramienta de los tests funcionales de regresión se encuentran varios casos:

- Mejor de los casos.

En el mejor de los casos el tiempo de ejecución que se ha obtenido ha sido de dos segundos, devolviendo instantáneamente el set de regresiones que se ha de ejecutar cuando se ha modificado una única función que solo ha implicado la ejecución de un subset.

- Caso promedio.

El caso promedio se ha dado entre 20 minutos y 25 minutos, puesto que estaban involucradas varias funciones y las dependencias de las mismas.

- Peor de los casos.

El peor de los casos obtenido ha sido de 55 minutos donde se han modificado varias funciones de distintos módulos y algunas dependencias de otros módulos.

7. Conclusiones y trabajo futuro

En este apartado se van a comentar las conclusiones finales a las que se han llegado tras realizar el proyecto al igual que también se van a comentar las posibles mejoras a realizar en un futuro.

7.1. Conclusiones finales

Inicialmente en el entorno previo al desarrollado, se llevaban a cabo las pruebas funcionales de los nuevos módulos de código por medio de un software bastante ineficiente con respecto al tiempo de ejecución y tiempo de respuesta de estas pruebas por lo que cada vez que se modificaba la más mínima parte del código del producto de la compañía se tenía necesariamente que ejecutar toda la suite de test de regresión independientemente de si los escenarios se correspondían o no a los necesarios para testear las líneas de código modificadas del producto.

También se partía de que cada vez que un usuario deseaba ejecutar los test, debía de crearse su correspondiente usuario de regresión en las bases de datos necesarias para llevar a cabo este proceso modificando manualmente un fichero donde se tienen almacenados todos los usuarios con sus datos en Bitbucket y además llevando las tareas como la de sincronización manualmente lo que en ocasiones resultaba laborioso.

Por todo ello se decidió crear una herramienta la cual nos realizase esta creación de usuario de regresión de forma automática por medio de una dashboard en la cual se rellenesen de forma obligatoria únicamente un formulario con dos campos: Unix id de usuario y máquina en la que trabaja; teniendo además como opciones avanzadas el formulario en el cual se pueden insertar también el nombre que se le quiere dar a este usuario en la/s bases de datos, el número de puerto y la base de datos en la que se va a crear por ejemplo.

Esta herramienta nos crea de forma automática el usuario en las bases de datos asignándole el alias y un puerto de forma inteligente evitando la sobrecarga.

Otro objetivo que se ha cumplido ha sido el de crear la herramienta la cual facilita el testeo del nuevo código del producto de la compañía que se haya modificado o creado. Esta herramienta ha cumplido con sus objetivos ya que ha permitido subdividir la suite de regresiones en subsets y sus dependencias de forma que cuando se modifique alguna parte del código del producto únicamente se ejecuten los subsets de pruebas de regresión implicados, evitando así tener que lanzar la suite entera por lo que se reduce considerablemente el tiempo de respuesta.

Finalmente la herramienta se ha puesto en producción únicamente en uno de los entorno y actualmente se viene usando en el mismo.

7.2. Trabajo futuro

Como trabajo futuro el cual se prevé abordar en los siguientes meses es aplicar la herramienta desarrollada para trabajar en conjunto con las regresiones dockerizadas como se mencionó en el capítulo de este proyecto dedicado a los objetivos abordables en el futuro.

Como se mencionó, el proyecto de dockerización está teniendo como principal objetivo convertir las partes integrales del sistema en grupos de trabajo autónomos que se distribuirán en una granja de servidores desplegada en nuestra infraestructura o en alguna nube, IaaS.

En este apartado los usuarios llevarían a cabo el inicio de una secuencia de instrucciones la cual nos crearía o reutilizaría las imágenes que ya se tienen y crearía un grupo de trabajo , se ejecutarían el subconjunto de regresiones necesarias para las pruebas del código y se obtendría el resultado de estos al final informando al usuario.

Además, se crearía una interfaz mas amigable para el usuario al igual que se seguirán con las labores de mantenimiento de las herramientas creadas.

Bibliografía

- [1] Amadeus SI. <https://amadeus.com/en>
- [2] Diagrama Gantt. <https://www.tomsplanner.es/?template=example>
- [3] Valgrind. <http://www.valgrind.org/>
- [4] Callgraph.
http://www.keil.com/support/man/docs/armlink/armlink_pge1362075422709.htm
- [5] Operf file suite: <http://oprofile.sourceforge.net/about/>
- [6] Operf profiler tutorial to generate callgraph:
<https://rndwww.nce.amadeus.net/confluence/pages/viewpage.action?spaceKey=C1&title=Operf+profiler>
- [7] Django. <https://www.djangoproject.com/>
- [8] Gprof. <https://sourceware.org/binutils/docs/gprof/>
- [9] Valgrind.
http://access.redhat.com/site/documentation/Red_Hat_Enterprise_Linux/
- [10] Oprofile. <http://oprofile.sourceforge.net/docs/>
- [11] Vim. <https://www.vim.org/>
- [12] Pyparser. <http://doc.pypy.org/en/latest/parser.html>
- [13] Cscope. <http://cscope.sourceforge.net/>
- [14] CppHeaderParser. <https://pypi.org/project/CppHeaderParser/>
- [15] LLVM. <https://github.com/llvm-mirror/llvm>
- [16] Clang. <https://github.com/llvm-mirror/clang>
- [17] Neo4J: <https://bbvaopen4u.com/es/actualidad/neo4j-que-es-y-para-que-sirve-una-base-de-datos-orientada-grafos>
- [18] Tutorial Clang.
http://rodin.uca.es/xmlui/bitstreamh/andle/104981/6912/clang_tutorial.pdf

[19] Running examples: <https://heejune.me/2016/08/09/wheres-my-llvmtosclangexample-binaries/>

[20] Proxy Squid.

<https://rndwww.nce.amadeus.net/confluence/pages/viewpage.action?spaceKey=~wherka&title=How+to+setup+your+own+http+proxy>

[21] LLVM Tutorial 1: A First Function.

<http://releases.llvm.org/2.6/docs/tutorial/JITTutorial1.html>

[22] PrintFunctionNames: <https://clang.llvm.org/docs/ClangPlugins.html> ,
<http://llvm.org/viewvc/llvm-project/cfe/trunk/examples/PrintFunctionNames/README.txt?view=markup>

[23] Build clang/examples/PrintFunctionNames:

<https://stackoverflow.com/questions/6265624/how-to-build-clang-examples-printfunctionnames>

[24] Printing function name (to retrieve namespace) from compiler output:
<https://stackoverflow.com/questions/47893828/printing-function-name-to-retrieve-namespace-from-compiler-output>

Glosario de términos

- Clang: Clang es un front end de compilador para los lenguajes de programación C, C++, Objective-C y Objective-C++.
- Django: es un framework para aplicaciones web gratuito y open source, escrito en Python. Es un WEB framework, un conjunto de componentes que te ayudan a desarrollar sitios web más fácil y rápidamente.
- Docker: es un proyecto de código abierto que automatiza el despliegue de aplicaciones dentro de contenedores de software, proporcionando una capa adicional de abstracción y automatización de Virtualización a nivel de sistema operativo en Linux.
- Doxygen: es un generador de documentación el cual se caracteriza por ser fácilmente adaptable, funciona en la mayoría de sistemas Unix así como en Windows y Mac OS X.
- Escenarios: son los posibles escenarios o casos los cuales testean el código del producto.
- JSON: (JavaScript Object Notation o Notación de Objetos de JavaScript) es un formato ligero de intercambio de datos.
- LLVM: es una infraestructura para desarrollar compiladores, escrita a su vez en el lenguaje de programación C++, que está diseñada para optimizar el tiempo de compilación, el tiempo de enlazado, el tiempo de ejecución y el tiempo en cualquier lenguaje de programación que el usuario quiera definir.
- Neo4J: es un software libre de Base de datos orientada a grafos.
- Oprofile: es una herramienta de supervisión de rendimiento que se ejecuta a lo largo de todo el sistema.
- Proxy: es un punto intermedio entre un ordenador conectado a Internet y el servidor que está accediendo.
- SFA (Software Framework Architecture): nombre del departamento en el que se ha desarrollado el proyecto.
- Squid: es un servidor proxy para web con caché.
- Test de regresión: son pruebas de software que intentan descubrir errores, carencias de funcionalidad, o divergencias funcionales con respecto al comportamiento esperado del software, causados por la realización de un cambio en el programa
- Usuario de regresión: es el usuario que se crea con permisos para poder ejecutar los tests de regresión del producto.
- Valgrind: es una herramienta de programación para depuración de memoria, detección de fuga de memoria y creación de perfiles.

Anexos

Solución al problema con Clang en la maquina IonInx48:

- ```
export PATH=/remote/users/kaxelrod/bin/vim74/bin:/opt/Python-2.7/bin:/data/ngddeldev/int/ngd_pythonpack/2-0-0-0/bin:/softntools/opt/Python-2.7/../../libxml2.7.7/bin:/data/softs/jre1.6.0_26/bin:/opt/devsup/platinum:/opt/devsup/mercuria:/bin:/projects/ngdcs/SI/linux/tools:/opt/devsup/platnum:/softntools/opt/Python-2.7/../../libxml/2.7.7/bin:/opt/devsup/cppcheck/latest:/opt/devsup/ocr:/opt/devsup/cmk/latest:/opt/devsup/bms/latest:/opt/devsup/mercurial/bin:/opt/devsup/ctm:/opt/gcc-4.3.2/bin:/data/ngddeldev/bin:/data/ngdcs/bin:/data/mwdeldev/bin:/data/softs/bin:/data/usr/bin:/data/softs/oracle/10.2.0.3//bin:/softntools/opt/Python-2.7/../../libxml/2.7.7/bin:/data/ngddeldev/int/ngd_softs/default/install/bin:/na/stools/openssl/bin/openssl:/opt/devsup/platinum:/opt/Python-2.7/bin:/data/ngddeldev/int/ngd_pythonpack/2-0-0-2/bin:/softntools/opt/Python-2.7/../../libxml/2.7.7/bin:/opt/devsup/cppcheck/latest:/opt/devsup/ocr:/opt/devsup/cmk/latest:/opt/devsup/ctm:/opt/gcc-4.3.2/bin:/na/stools/git/bin:/data/ngddeldev/bin:/data/ngdcs/bin:/data/mwdeldev/bin:/data/softs/bin:/data/usr/bin:/data/softs/oracle/10.2.0.3//bin:/softntools/opt/Python-2.7/../../libxml/2.7.7/bin:/usr/bin:/bin:/usr/sbin:/sbin/x86_64-2.6.32-v4.0.11/lib/libclang.so.5.0
```
- ```
export LD_LIBRARY_PATH=/x86_64-2.6.32-v4.0.11/lib:/x86_64-2.6.32-v4.0.11/lib/libc.so.6:/softntools/opt/Python-2.7/../../libxslt/1.1.26/lib:/remote/tools/Linux/2.6/oracle/products/10.2.0.4/lib:/softntools/opt/Python-2.7/lib:/softntools/opt/Python-2.7/lib/python2.7/lib-dynload:/data/usr/lib:/data/softs/oracle/10.2.0.4/lib:/softntools/opt/Python-2.7/lib:/softntools/opt/Python-2.7/lib/python2.7/lib/python2.7/lib-dynload:/data/softs/hdf5-1.8.13/lib:/data/softs/hdf5-1.8.13/lib:
```

