

**AUTOMATIZACIÓN DE PRUEBAS FUNCIONALES PARA LA APLICACIÓN WEB
FONDO DE PENSIONES OBLIGATORIAS DE LA GESTORA PUBLICA DE BOLIVIA**

Danna Michelle Arévalo Hernández

Facultad de Ingeniería y Ciencias Básicas, Universidad Politécnico GranColombiano

Trabajo Opción de Grado

Tutora Isabel Mahecha

Mayo 23 del 2022

Universitaria Politécnico GranColombiano

Sede Bogotá

TABLA DE CONTENIDO

| | |
|---|----|
| Capítulo 1. INTRODUCCIÓN | 14 |
| 1.1 Contexto de la empresa | 15 |
| 1.2 Planteamiento del problema | 16 |
| 1.3 Objetivos | 16 |
| 1.4 Justificación | 17 |
| 1.5 Alcance | 17 |
| Capítulo 2. MARCO TEORICO | 19 |
| 2.1 Calidad..... | 19 |
| 2.2 Calidad de software..... | 19 |
| 2.3 Normativa existente de calidad..... | 19 |
| 2.4 Automatización de pruebas..... | 20 |
| 2.5 Herramientas..... | 21 |
| 2.6 Introducción a las pruebas de software..... | 23 |
| 2.7 Niveles de Pruebas De Software..... | 23 |
| 3.3 Conclusiones..... | 24 |
| Capítulo 3. Implementación | 25 |
| 3.1 Introducción a Selenium..... | 25 |
| 3.1.1. Selenium 1.0..... | 25 |
| 3.1.2. Componentes RC..... | 25 |
| 3.1.3. Selenium 2.0 (Selenium WebDriver) | 25 |
| 3.2 Introducción implementación..... | 26 |
| 3.3 Instalación Java | 27 |
| 3.4 Instalación IntelliJ (IDE) | 29 |
| 3.5 Maven | 29 |
| 3.6 ChromeDriver | 36 |
| 3.7 Data Transfer Object (DTO) | 42 |
| 3.8 Page Object Model (POM) | 44 |

| | |
|--|------------|
| 3.9 TestNG | 47 |
| 3.10 Anotaciones Comunes..... | 48 |
| 3.11 Implementación Test..... | 49 |
| 3.12. Interfaz WebDriver (Localización de Elementos Web) | 52 |
| Capítulo 4. Métodos para localizar elementos en la web de la clase By | 54 |
| Capítulo 5. Metodología de desarrollo de software | 54 |
| Capítulo 6. Resultados | 55 |
| 6.1 Introducción | 55 |
| 6.2 Funcionalidad Registro Empleador..... | 55 |
| 6.3 Funcionalidad Registro Consultor..... | 71 |
| 6.4 Funcionalidad Registro Desclasificación Asegurados Militares..... | 88 |
| 6.5 Funcionalidad Cierre de tramos laborales de asegurados fallecidos..... | 102 |
| 6.6 Funcionalidad Registro Asegurados..... | 111 |
| 6.7 Sucursales empleador..... | 121 |
| 6.8 Funcionalidad Cierre de tramos laborales de entidades públicas y privadas..... | 122 |
| 6.9 Análisis de los resultados..... | 125 |
| Capítulo 7. Referencias..... | 126 |
| Capítulo 8. Trabajo futuro | 127 |

INDICE DE FIGURAS

| | |
|---|----|
| Ilustración 1CMD. Elaboración Propia. | 26 |
| Ilustración 2. Variables de entorno. Elaboración Propia. | 27 |
| Ilustración 3. Variables de entorno. Elaboración Propia. | 27 |
| Ilustración 4: Descarga IntelliJ IDEA. Elaboración Propia. | 28 |
| Ilustración 5: Pom.xml. Elaboración Propia. | 29 |
| Ilustración 6: Descarga Maven. Elaboración Propia. | 30 |
| Ilustración 7: Variables de entorno. Elaboración Propia. | 31 |
| Ilustración 8: Editar Variable Sistema. Elaboración Propia. | 31 |
| Ilustración 9: Símbolo del Sistema. Elaboración Propia. | 32 |
| Ilustración 10: Settings. Elaboración Propia. | 32 |
| Ilustración 11. Ajustes Maven IntelliJ. Elaboración Propia. | 33 |
| Ilustración 12. Dependencias POM. Elaboración Propia. | 33 |
| Ilustración 13. Carpetas Proyectos. Elaboración Propia. | 34 |
| Ilustración 14. Configuración ChromeDriver. Elaboración Propia. | 35 |
| Ilustración 15. Versión Chrome. Elaboración Propia. | 36 |
| Ilustración 16. Chrome Driver. Elaboración Propia. | 37 |
| Ilustración 17. Versiones Chrome Driver. Elaboración Propia. | 37 |
| Ilustración 18. Carpeta drivers. Elaboración Propia. | 38 |
| Ilustración 19. Carpeta Proyecto. Elaboración Propia. | 38 |
| Ilustración 20. Generación automática métodos GET y SET. Elaboración Propia. | 39 |
| Ilustración 21. Métodos Getter and Setter. Elaboración Propia. | 40 |

| | |
|---|----|
| Ilustración 22. Métodos GET y SET. Elaboración Propia..... | 40 |
| Ilustración 23. Clases Proyecto. Elaboración Propia..... | 41 |
| Ilustración 24. Clase DTO. Elaboración Propia. | 43 |
| Ilustración 25. Clase DTO. Elaboración Propia. | 44 |
| Ilustración 26. Clase Page. Elaboración Propia. | 45 |
| Ilustración 27. Clase Page. Elaboración Propia. | 45 |
| Ilustración 28. Método Clase Page. Elaboración Propia. | 46 |
| Ilustración 29. Dependencias. Elaboración Propia. | 47 |
| Ilustración 30. TestNG. Elaboración Propia. | 48 |
| Ilustración 31. Nota. Adaptado de Programmerclick.com. 2022. Notas de estudio de selenio- marco de prueba de unidad- | |
| | Te |
| stNG - programador clic. Disponible en: < https://programmerclick.com/article/2681954615/ > | 49 |
| Ilustración 32. Clase Test. Elaboración Propia. | 49 |
| Ilustración 33. Clase Test. Elaboración Propia. | 49 |
| Ilustración 34. Método Clase Test. Elaboración Propia. | 50 |
| Ilustración 35. Método Clase Test. Elaboración Propia. | 50 |
| Ilustración 36. Método Clase Test. Elaboración Propia. | 51 |
| Ilustración 37. AfterMethod Clase Test. Elaboración Propia. | 51 |
| Ilustración 38. WebElement..... | 53 |
| Ilustración 39. Web Element Clase By..... | 54 |
| Ilustración 40. Funcionalidad Empleador. Elaboración Propia. | 56 |
| Ilustración 41. DPC Adicionar Solicitud Desclasificación Militar. Elaboración Propia..... | 57 |

| | |
|--|----|
| Ilustración 42. DPC Adicionar Solicitud Desclasificación Militar. Elaboración Propia. | 58 |
| Ilustración 43. DPC Consulta Desclasificación Asegurado Militar. Elaboración Propia. | 59 |
| Ilustración 44. DPC Filtros Solicitud Desclasificación de Asegurados Militares. Elaboración Propia. | 60 |
| Ilustración 45. Generar certificación militar. Elaboración Propia. | 61 |
| Ilustración 46. Script Before. Elaboración Propia. | 62 |
| Ilustración 47. Script After. Elaboración Propia. | 62 |
| Ilustración 48. Carpetas Proyecto. Elaboración Propia. | 62 |
| Ilustración 49. Clase Dto Empleador. Elaboración Propia. | 63 |
| Ilustración 50. Clase Dto Empleador. Elaboración Propia. | 64 |
| Ilustración 51. Clase Page Empleador. Elaboración Propia. | 65 |
| Ilustración 52. Clase Page Empleador. Elaboración Propia. | 65 |
| Ilustración 53. Método Clase Page Empleador. Elaboración Propia. | 66 |
| Ilustración 54. Clase Test Adicionar Solicitud Empleador. Elaboración Propia. | 67 |
| Ilustración 55. Método Clase Test Adicionar Solicitud Empleador. Elaboración Propia. | 67 |
| Ilustración 56. Método Clase Test Adicionar Solicitud Empleador. Elaboración Propia. | 68 |
| Ilustración 57. Método Clase Test Adicionar Solicitud Empleador. Elaboración Propia. | 68 |
| Ilustración 58. AfterMethod Clase Test Adicionar Solicitud Empleador. Elaboración Propia. | 69 |
| Ilustración 59. Comando Ejecución de Pruebas en Grupo. Elaboración Propia. | 69 |
| Ilustración 60. Carpeta Reporte Pruebas Selenium. Elaboración Propia. | 70 |
| Ilustración 61. Reporte de ejecución de Pruebas Desclasificación de Asegurados Militares. Elaboración Propia. | 70 |

| | |
|---|----|
| Ilustración 62. Reporte de ejecución de Pruebas Desclasificación de Asegurados Militares. Elaboración Propia. | 71 |
| Ilustración 63. Funcionalidad Formulario Registro de Solicitud de Consultores. Elaboración Propia. | 72 |
| Ilustración 64. Localizador id. Elaboración Propia. | 73 |
| Ilustración 65. DPC Adicionar Solicitud de Registro de Consultor. Elaboración Propia. | 73 |
| Ilustración 66. DPC Adicionar Solicitud de Registro de Consultor. Elaboración Propia. | 74 |
| Ilustración 67. DPC Adicionar Solicitud de Registro de Consultor. Elaboración Propia. | 74 |
| Ilustración 68. DPC Adicionar Solicitud de Registro de Consultor. Elaboración Propia. | 75 |
| Ilustración 69. DPC Filtros Solicitud de Registro de Consultor. Elaboración Propia. | 76 |
| Ilustración 70. DPC Escenario Uno Histórico Solicitud de Registro de Consultor. Elaboración Propia. | 77 |
| Ilustración 71. DPC Escenario Dos Histórico Solicitud de Registro de Consultor. Elaboración Propia. | 77 |
| Ilustración 72. DPC Rechazar Solicitud de Registro de Consultor. Elaboración Propia. | 77 |
| Ilustración 73. DPC Validar Solicitud de Registro de Consultor. Elaboración Propia. | 78 |
| Ilustración 74. DPC Editar Solicitud de Registro de Consultor. Elaboración Propia. | 78 |
| Ilustración 75. DPC Escenario Uno Consulta Solicitud de Registro de Consultor. Elaboración Propia. | 79 |
| Ilustración 76. DPC Escenario Dos Consulta Solicitud de Registro de Consultor. Elaboración Propia. | 79 |
| Ilustración 77. Script Registro Consultor. Elaboración Propia. | 79 |
| Ilustración 78. Delete Script Registro Consultor. Elaboración Propia. | 80 |

| | |
|--|----|
| Ilustración 79. Script Registro Consultor. Elaboración Propia. | 80 |
| Ilustración 80. Script Registro Consultor. Elaboración Propia. | 80 |
| Ilustración 81. Carpeta Funcionalidad Solicitud Consultor. Elaboración Propia. | 81 |
| Ilustración 82. Clase Dto Adicionar Registro Consultor. Elaboración Propia. | 81 |
| Ilustración 83. Clase Dto Adicionar Registro Consultor. Elaboración Propia. | 82 |
| Ilustración 84. Clase Page Adicionar Registro Consultor. Elaboración Propia. | 82 |
| Ilustración 85. Método Clase Page Adicionar Registro Consultor. Elaboración Propia. | 83 |
| Ilustración 86. Método Clase Page Adicionar Registro Consultor. Elaboración Propia. | 83 |
| Ilustración 87. Clase Test Adicionar Consultor. Elaboración Propia. | 84 |
| Ilustración 88. Clase Test Adicionar Consultor. Elaboración Propia. | 84 |
| Ilustración 89. Método Clase Test Adicionar Consultor. Elaboración Propia. | 85 |
| Ilustración 90. Método Clase Test Adicionar Consultor. Elaboración Propia. | 85 |
| Ilustración 91. AfterMethod Clase Test Adicionar Consultor. Elaboración Propia. | 86 |
| Ilustración 92. Comando ejecución de pruebas automáticas en grupo. Elaboración Propia. ... | 86 |
| Ilustración 93. Reportes Pruebas Adicionar Solicitud de Consultor. Elaboración Propia. | 87 |
| Ilustración 94. Reportes Pruebas Aprobar Solicitud de Consultor. Elaboración Propia. | 87 |
| Ilustración 95. Reportes Pruebas Consultar Solicitud de Consultor. Elaboración Propia. | 87 |
| Ilustración 96. Reportes Pruebas Filtros Solicitud de Consultor. Elaboración Propia. | 88 |
| Ilustración 97. Funcionalidad Desclasificación Asegurados Militares. Elaboración Propia. ... | 88 |
| Ilustración 98. Verificación Localizadores id's. Elaboración Propia. | 89 |
| Ilustración 99. DPC Adicionar Solicitud de Desclasificación de Asegurados Miliars. Elaboración Propia. | 90 |

| | |
|---|----|
| Ilustración 100. DPC Consultar Solicitud de Desclasificación de Asegurados Militares. Elaboración Propia..... | 91 |
| Ilustración 101. DPC Filtros Desclasificación de Asegurados Militares. Elaboración Propia. | 92 |
| Ilustración 102. DPC Generar Certificación de Militar. Elaboración Propia. | 93 |
| Ilustración 103. Carpetas Funcionalidad Desclasificación. Elaboración Propia..... | 94 |
| Ilustración 104. Carpetas Funcionalidad Desclasificación. Elaboración Propia..... | 94 |
| Ilustración 105. Carpetas Funcionalidad Desclasificación. Elaboración Propia..... | 94 |
| Ilustración 106. Test Adicionar Solicitud de Desclasificación de Asegurado Militar. Elaboración Propia..... | 95 |
| Ilustración 107. Test Adicionar Solicitud de Desclasificación de Asegurado Militar. Elaboración Propia..... | 95 |
| Ilustración 108. Ejecución de Script dentro del Test Adicionar Solicitud de Desclasificación de Asegurado Militar. Elaboración Propia. | 96 |
| Ilustración 109. Método del Test Adicionar Solicitud de Desclasificación de Asegurado Militar. Elaboración Propia. | 96 |
| Ilustración 110. Método del Test Adicionar Solicitud de Desclasificación de Asegurado Militar. Elaboración Propia. | 96 |
| Ilustración 111. Método del Test Adicionar Solicitud de Desclasificación de Asegurado Militar. Elaboración Propia. | 97 |
| Ilustración 112. AfterMethod Encargado de Cerrar Venta Navegador del Test Adicionar Solicitud de Desclasificación de Asegurado Militar. Elaboración Propia. | 97 |
| Ilustración 113. Page Adicionar Solicitud de Desclasificación de Asegurado Militar. Elaboración Propia..... | 98 |

| | |
|--|-----|
| Ilustración 114. Page Adicionar Solicitud de Desclasificación de Asegurado Militar. Elaboración Propia..... | 98 |
| Ilustración 115. Método del Page Adicionar Solicitud de Desclasificación de Asegurado Militar. Elaboración Propia. | 99 |
| Ilustración 116. Método del Page Adicionar Solicitud de Desclasificación de Asegurado Militar. Elaboración Propia. | 99 |
| Ilustración 117. Clase Dto Adicionar Solicitud de Desclasificación de Asegurado Militar. Elaboración Propia..... | 100 |
| Ilustración 118. Comando Ejecución Pruebas Automáticas en Grupo. Elaboración Propia. . | 101 |
| Ilustración 119. EPC Reporte de Pruebas de Desclasificación de Asegurados Militares. Elaboración Propia..... | 101 |
| Ilustración 120. Funcionalidad Cierre tramos laborales de los asegurados fallecidos. Elaboración Propia..... | 103 |
| Ilustración 121. DPC Filtro Asegurado Fallecido. Elaboración Propia. | 104 |
| Ilustración 122. Filtro Asegurado Fallecido por CUA. Elaboración Propia..... | 104 |
| Ilustración 123. DPC Generar Cierre de tramos laboral asegurado fallecido manual. Elaboración Propia..... | 105 |
| Ilustración 124. Test Filtro Asegurado Fallecido. Elaboración Propia. | 105 |
| Ilustración 125. Test Filtro Asegurado Fallecido. Elaboración Propia. | 106 |
| Ilustración 126. Método de la clase Test Filtro Asegurado Fallecido. Elaboración Propia. .. | 106 |
| Ilustración 127. Método de la clase Test Filtro Asegurado Fallecido. Elaboración Propia. .. | 107 |
| Ilustración 128. AfterMethod de la clase Test Filtro Asegurado Fallecido encargado de cerrar la ventana del navegador. Elaboración Propia..... | 107 |

| | |
|--|-----|
| Ilustración 129. Page Filtro Asegurado Fallecido. Elaboración Propia. | 108 |
| Ilustración 130. Page Filtro Asegurado Fallecido. Elaboración Propia. | 108 |
| Ilustración 131. Método de la clase Page Filtro Asegurado Fallecido. Elaboración Propia...108 | |
| Ilustración 132. Método de la clase Page Filtro Asegurado Fallecido. Elaboración Propia...109 | |
| Ilustración 133. Dto Filtro Asegurado Fallecido. Elaboración Propia. | 109 |
| Ilustración 134. Comando Ejecución Pruebas automáticas en Grupo. Elaboración Propia. .. | 110 |
| Ilustración 135. EPC Reporte de pruebas automáticas función cierre de tramos laborales de asegurados fallecidos. Elaboración Propia. | 110 |
| Ilustración 136. Crear Solicitud de Vinculación. Elaboración Propia..... | 111 |
| Ilustración 137. Localizadores por id. Elaboración Propia..... | 112 |
| Ilustración 138. Gestionar Solicitudes de Vinculación. Elaboración Propia. | 112 |
| Ilustración 139. DPC Crear Vinculación. Elaboración Propia..... | 114 |
| Ilustración 140. DPC Rechazar Solicitud de Vinculación. Elaboración Propia. | 114 |
| Ilustración 141. Test Crear Vinculación. Elaboración Propia. | 115 |
| Ilustración 142. Test Crear Vinculación. Elaboración Propia. | 115 |
| Ilustración 143. Test Crear Vinculación. Elaboración Propia. | 115 |
| Ilustración 144. Test Crear Vinculación. Elaboración Propia. | 116 |
| Ilustración 145. Test Crear Vinculación. Elaboración Propia. | 116 |
| Ilustración 146. Page Crear Vinculación. Elaboración Propia..... | 117 |
| Ilustración 147. Método del Page Crear Vinculación. Elaboración Propia. | 117 |
| Ilustración 148. Método del Page Crear Vinculación. Elaboración Propia. | 118 |
| Ilustración 149. Método del Page Crear Vinculación. Elaboración Propia. | 118 |
| Ilustración 150. Método del Page Crear Vinculación. Elaboración Propia. | 119 |

| | |
|---|-----|
| Ilustración 151. Método del Page Crear Vinculación. Elaboración Propia. | 119 |
| Ilustración 152. Dto Crear Vinculación. Elaboración Propia. | 120 |
| Ilustración 153. EPC Reporte Pruebas Vinculación de Asegurados. Elaboración Propia. | 120 |
| Ilustración 154. Funcionalidad Sucursales Empleador. Elaboración Propia. | 121 |
| Ilustración 155. Funcionalidad Sucursales Empleador. Elaboración Propia. | 121 |
| Ilustración 156. Funcionalidad Sucursales Empleador. Elaboración Propia. | 122 |
| Ilustración 157. Funcionalidad Cierre de tramos laborales de entidades públicas y privadas. Elaboración Propia. | 122 |
| Ilustración 158. Funcionalidad Cierre de tramos laborales de entidades públicas y privadas. Elaboración Propia. | 123 |
| Ilustración 159. Ejecución Pruebas a través de Postman Funcionalidad Cierre de tramos laborales de entidades públicas y privadas. Elaboración Propia. | 123 |
| Ilustración 160. Ejecución Pruebas a través de Postman Funcionalidad Cierre de tramos laborales de entidades públicas y privadas. Elaboración Propia. | 124 |

Capítulo 1. INTRODUCCIÓN

Producir un Software eficiente y de buena calidad es algo muy importante para las empresas hoy en día, toda empresa en la industria de software buscan producir software de buena calidad, con estándares altos, considerando esto como una exigencia dentro del proceso de desarrollo; en consecuencia de esto la automatización de pruebas es un paso fundamental para mejorar la calidad del desarrollo de software que se esté haciendo, la automatización se puede presentar en diferentes niveles de pruebas desde pruebas unitarias hasta pruebas de aceptación, en este proyecto se incluye el proceso de automatización de pruebas funcionales donde se simula la interacción entre el cliente final con la aplicación para poder verificar que su funcionamiento sea el correcto en la interfaz web. Hoy en día la herramienta de automatización de pruebas funcionales de Selenium se ha vuelto demasiado importante, ya que nos permite hacer toda la interacción del usuario con el aplicativo desde abrir el navegador, ejecución de scripts, ingreso a la URL del sitio, diligenciar usuarios, contraseñas y diferentes datos que este solicitando el aplicativo hasta terminar con un determinado proceso que se quiera realizar en diferentes casos. En el presente proyecto de opción de grado se describen los conceptos básicos relacionado con la automatización de pruebas y se realiza toda la fase de pruebas en el proyecto de Software Pensiones Obligatorias por parte de la empresa Heinsohn Business Technology, donde se utiliza la herramienta de Selenium WebDriver como se nombró anteriormente. Esta herramienta de automatización permite programar pruebas en un lenguaje de alto nivel, en este caso Java haciendo uso del API que proporciona. Gracias a esto dentro de Selenium WebDriver podemos simular la interacción entre un usuario, en este caso el cliente sobre la aplicación de Pensiones Obligatorias. En este proyecto se evidencia y se pone en práctica todos los conocimientos aprendidos y se entrega toda la parte de automatización de pruebas.

El mercado va cambiando continuamente y las exigencias de un software de calidad cada día son más altas, es por ellos que las pruebas automáticas nos ayudan a verificar procesos repetitivos para poder verificar su correcto funcionamiento.

Con el fin de garantizar el buen funcionamiento del software en el presente proyecto se muestran los antecedentes, la relevancia que tiene las pruebas automáticas hoy en día a la hora de entregar un software al cliente, los objetivos, se da introducción a Selenium WebDriver con sus Interfaces, métodos y toda la parte de la configuración del entorno de pruebas para posteriormente llevar a cabo la ejecución de pruebas y obtener un reporte de los resultados de pruebas.

1.1. Contexto de la empresa

Heinsohn Business Technology es una empresa Colombiana con más de 40 años ofreciendo soluciones empresariales a través de la tecnología, los fundadores de Heinsohn Johan y Miguel Sáenz crean la empresa en 1977, donde iba a ser la mayor empresa de trayectoria en Colombia, en 1983 Heinsohn lanza al mercado el primer software ERP llamado Aurora Nomina, en el 2006 inician alianzas con SAP Business One, en el 2009 Heinsohn desarrolla e implementa el RUNT en Colombia, en el 2015 ganan el premio #1 al Partner en Colombia y el #4 en el mundo con SAP Business One, en el 2017 MinTIC y Procolombia hacen reconocimiento a la empresa como exportadora mayor de actividad comercial en el año, para el 2017 recibieron valoración CMMI SVC nivel 5 1.3; lograron la calificación más alta en proceso de servicios, en el 2020 se certificó en CMMI 2.0; Heinsohn se convirtió en la primera empresa Colombiana en recibir certificación CMMI 2.0 DEV y SVC por su estándares de calidad. “Esta compañía de tecnología está comprometida con la calidad e innovación permanente y prueba de ello son las valoraciones CMMI Nivel 5 versión 2.0, pues Heinsohn es una de las primeras

empresas del mundo en recibir la valoración más alta en la calidad de los procesos de desarrollo de software, de igual manera sus servicios en TI están certificados por su efectividad y alcance con la valoración CMMI Nivel 5 que marcan un gran diferencial y un valor agregado para sus clientes, el Sistema de Gestión de Seguridad y Salud en el Trabajo de la compañía es avalado por la certificación OSHAS 18001, además contamos con la aprobación del Sistema de Gestión de Calidad ISO 9001 que ayuda al fortalecimiento de los productos y servicios.” (Heinsohn).

1.2. Planteamiento del problema

Actualmente la empresa Heinsohn está realizando el proyecto Pensiones Obligatorias, según los estatutos y la legislación que entró en vigor con el nuevo gobierno para Bolivia. El desarrollo web cuenta con varios módulos, durante este proyecto se trabajaron en las funcionalidades implementadas del módulo de Personas. Este Modulo cuenta con varias acciones como crear, modificar, eliminar y consultar el histórico de un Asegurado o de un Empleador, existen dos tipos de personas que se pueden crear, la primera de ellas es la Persona Natural (Asegurado) donde se puede afiliarse al fondo de pensiones y el segundo tipo de persona que se puede crear es una Persona Jurídica (Empleador) donde pueden llegar a tener contratos con dichos afiliados. A lo largo de este proyecto se crearon nuevas funcionalidades en la plataforma web que tiene que ver con los empleadores (Las personas jurídicas) del cual se tuvo la responsabilidad del Control de Calidad del Producto de Software que se implementó antes de salir a producción, la misión que se abarcó fue asegurar que el software que se entregó funcione de acuerdo a los requisitos solicitados por el cliente, mirando que las interfaces (desarrolladas utilizando el Framework Angular) funcionen de la forma esperada, detectando así cualquier defecto antes de la entrega al cliente. Se construyó todo el plan testing, diseño de casos de pruebas, construcción de pruebas

programadas en Java utilizando el IDE de IntelliJ, gestión de los ambientes de prueba, generación de datos de prueba utilizando scripts para los motores de base de datos Oracle como Postgres, pruebas automatizadas utilizando la herramienta de Selenium WebDriver, ejecución y documentación de pruebas (evidencia), reportes y seguimiento a cualquier incidente encontrado, es importante resaltar que todas estas pruebas automatizadas corren sobre el servidor web de WildFly.

1.3. Objetivos

OBJETIVO PRINCIPAL

Implementar el proceso de pruebas funcionales automatizadas para la empresa Heinsohn Business Technology en el aplicativo Heinsohn Pensiones, para las funcionalidades que se trabajaran dentro de este proyecto.

OBJETIVOS SECUNDARIOS

Los objetivos secundarios que me permitirán alcanzar el objetivo principal son los siguientes:

1. Diseñar el ciclo completo de automatización de pruebas desde el diseño de pruebas de calidad, generación de scripts de datos para pruebas, generación de código, ejecución de pruebas y reportes de ejecución de pruebas automáticas.
2. Desarrollo de código para la automatización de acciones dentro de las pruebas funcionales.
3. Ejecución de pruebas automatizadas con Selenium de las funcionalidades Registro Empleador, Registro Consultor, Registro Desclasificación Asegurados Militares, Cierre de tramos laborales de asegurados fallecidos, Registro Asegurados, Sucursales empleador y Cierre de tramos laborales de entidades públicas y privadas.

1.4. Justificación

Las pruebas de software son muy importantes y necesarias para poder detectar errores que se puedan llegar a tener a la hora de desarrollar un aplicativo y a su vez prueba que cumpla con todos los requisitos y estándares solicitados por el cliente final. Los retos que se presentan hoy en día en las empresas es ofrecer un software de buena calidad, la empresa Heinsohn aplica los conocimientos en ingeniería de software, metodologías, modelos de calidad para la industria de software, se aplica conocimientos de lenguaje de consulta de Base de Datos, herramientas de ejecución, seguimiento de testing y a su vez la codificación, pruebas automatizadas, pruebas de rendimiento y documentación de pruebas utilizando la herramienta Selenium WebDriver. Todo esto es muy importante en el mundo laboral, para poder de esta forma entregar un software de buena calidad y resaltar como empresa frente a otras existentes|.

1.5. Alcance

Se incluye el plan de pruebas asignado para todas las nuevas funcionalidades que se implementaran a lo largo de los meses de febrero y julio, cubriendo toda la parte del diseño y ejecución de las pruebas por cada nueva funcionalidad brindando de esta manera un aplicativo de alta calidad para los clientes.

Capítulo 2. MARCO TEORICO

2.1. Calidad

La calidad se puede definir como la capacidad que tiene un objeto en particular de cumplir las necesidades o los requerimientos solicitados.

- Según Parasuraman, B. Zeithaml y L. Berry (1985, 1988) entendieron la calidad como aquella discrepancia existente entre lo esperado y lo percibido.
- Según la norma NC/ ISO 9000 2005 la calidad es entendida como el grado en el que un conjunto de características inherentes cumple con los requisitos.

2.2. *Calidad de software*

Cuando hablamos de calidad de software podemos recurrir a las siguientes definiciones formales:

- El estándar ISO 9000: 2000 define la calidad de software como “Grado en que un conjunto de características inherentes cumple con unos requerimientos”.
- Según IEEE STD 610.1990. IEEE Standard Glossary of Software Engineering Terminology define “La calidad del software es el grado con el que un sistema, componente o proceso cumple los requerimientos especificados y las necesidades o expectativas del cliente o usuario”.

2.3. Normativa existente de calidad

Según la norma ISO/IEC 25010:2011 define 8 características que definen la calidad de un producto de software:

- Funcionalidad
- Eficiencia

- Compatibilidad
- Usabilidad
- Confiabilidad
- Seguridad
- Mantenibilidad
- Portabilidad

2.4. Automatización de Pruebas

Al automatizar pruebas de software el objetivo de ellas es hacer más simple el trabajo repetitivo haciéndolo mucho más efectivo, de acuerdo con esto:

- Según Max Rehkopf las pruebas automatizadas consisten en la aplicación de herramientas de software para automatizar el proceso manual de revisión y validación de un producto de software que lleva a cabo una persona.
- La contaduría general de la nación define la automatización de pruebas como actividad por la que se garantiza que el sistema verifique su correcto funcionamiento de manera automática.

Dependiendo del objetivo con el cual se realicen las pruebas se puede clasificar de la siguiente manera:

- ✓ Pruebas Unitarias
- ✓ Pruebas de Integración
- ✓ Pruebas funcionales
- ✓ Pruebas de sistema
- ✓ Pruebas de aceptación

2.5. Herramientas

Actualmente existen utilidades que facilitan el proceso de pruebas de software para aplicaciones web, dichas herramientas nos permiten poder detectar cualquier error y garantizar que el producto cumple con todos los requerimientos solicitados por el cliente, estas herramientas de testing están a lo largo del proyecto, garantizando que se cumplan todas las fases del ciclo de prueba. Estas herramientas nacen para poder agilizar el proceso de pruebas, está claro que hoy en día las pruebas automáticas gastan mucho menos tiempo que una persona que pruebe manualmente, gracias a que se pueden probar bastantes escenarios de forma automática, también cabe resaltar que cuando se hace una actualización de software es necesario que las pruebas se ejecuten diariamente para verificar que todos los procesos funcionen correctamente y así de esta forma verificar que no se afectó ninguna de las funcionalidades del aplicativo; en un caso como este una persona que este realizando estas pruebas manualmente se va a gastar el doble de tiempo que lo que gastaría las pruebas automatizadas, por ello se hace necesario la automatización de pruebas, estas nos brindan resultados más precisos ahorrando de esta manera tiempo y brindando confiabilidad para el cliente. Estas herramientas nos permiten realizar los casos de prueba que se requieran en tareas repetitivas sin necesidad de intervención manual una vez que ya están realizadas.

Según Rodríguez, C. (2022, 19 abril) Históricamente, la automatización surgió para reducir el esfuerzo humano requerido en actividades que podrían ser replicadas por un sistema o máquina programable. Al automatizar pruebas de software se persigue el objetivo de simplificar el trabajo dispendioso, repetitivo o complejo, haciéndolo efectivo y más productivo. De esta manera, es posible ahorrar energía, tiempo y costos, al tiempo que libera a las personas para que se concentren en otras tareas.

Ya que se dio una pequeña introducción de que tan útiles son las herramientas que hay actualmente en el mundo de pruebas de software, se va a describir la herramienta que se utilizara en este proyecto para realizar las pruebas automatizadas.

Herramienta Selenium WebDriver es un entorno de pruebas de software de aplicaciones en la web, “WebDriver utiliza los API de automatización del navegador proporcionadas por los proveedores de navegadores para controlar el navegador y ejecutar pruebas. Esto es como si un usuario real estuviera operando el navegador. Dado que WebDriver no requiere que su API se compile con el código de la aplicación, no es intrusivo. Por lo tanto, está probando la misma aplicación que impulsa en vivo” (Selenium).

Según Gilberto Sánchez hay 4 componentes de Selenium que son:

1. Selenium IDE: es un entorno de desarrollo integrado para scripts de Selenium. Se implementa como una extensión de Firefox y permite grabar, editar y depurar pruebas. Este IDE incluye todo el Selenium Core, que permite grabar y reproducir de forma fácil las pruebas en el entorno real en el cual se ejecutarán. Se recomienda su uso para prototipos de pruebas, debido a que no es capaz de generar ciclos ni condiciones.
2. Selenium RC (Remote Control): es una herramienta para automatizar pruebas de interfaz de usuario (UI) de aplicaciones web. Consta de dos componentes: a) un servidor que actúa como proxy para controlar e interactuar con un navegador web. b) bibliotecas para crear programas para el servidor usando una amplia gama de lenguajes de programación. Fue la herramienta original de Selenium para pruebas web pero a partir de la versión 2 fue integrado con WebDriver y se prefiere usar este último.

3. Selenium WebDriver: también es una herramienta para automatizar pruebas UI de aplicaciones web, pero implementa un enfoque más moderno y estable que Selenium RC. WebDriver, a diferencia de RC no utiliza un middleware sino que controla el navegador comunicándose directamente con él.
4. Selenium Grid: se especializan en ejecutar múltiples pruebas a través de diferentes navegadores, sistemas operativo y máquinas. Puede conectarse con Selenium Remote especificando el navegador, la versión del navegador y el sistema operativo que desee. Hay dos elementos principales: hub y nodos.

2.6.Introducción a las pruebas de software

Las pruebas de software son de suma importancia para verifica y validar el funcionamiento de las aplicaciones de software, para poder garantizar que el producto que se está entregando sea de alta calidad, existen dos tipos de pruebas; pruebas manuales y pruebas automatizadas.

Según IBM la prueba de software es el proceso de evaluación y verificación de un producto o aplicación de software para saber si hace lo que se supone que debe hacer.

2.7.Niveles de Pruebas De Software

Existen diversas pruebas dentro del transcurso del desarrollo, en diferentes módulos y parte del aplicativo de software, según se definan los casos de prueba y lo que se quiera realizar se puede clasificar las pruebas de software como:

- ✓ Pruebas Unitarias: Estas verifican de forma independiente todas las funciones y los métodos de las clases o de los componentes de los módulos.

- ✓ Pruebas de integración: Verifica que todos los módulos en el aplicativo o los componentes funcionen en conjunto.
- ✓ Pruebas de aceptación: Estas verifican que todo el sistema del aplicativo funcione según lo acordado.
- ✓ Pruebas Funcionales: Se llevan a cabo para comprobar las funcionalidades críticas y la usabilidad, se valida todo el aplicativo con respecto a las especificaciones o los requerimientos dados por el cliente y especificados en los casos de uso.
- ✓ Pruebas de rendimiento: Se llevan a cabo pruebas de carga para evaluar el rendimiento de carga reales del aplicativo.
- ✓ Pruebas de regresión: Verifican distintos escenarios que funcionen continuamente como debería ser.
- ✓ Pruebas de usabilidad: Verifica si el funcionamiento es muy bueno al momento de que un cliente intente hacer una terminada tarea.

2.8.Conclusiones

Las pruebas de software ayudan a prevenir errores y mejorar el rendimiento de una aplicación de software, por ello es fundamental la elaboración y ejecución de estas para poder ofrecer al cliente calidad, seguridad y rendimiento. Gracias a las pruebas de software podemos identificar todo tipo de errores previamente y así evitar la pérdida del negocio o las diferentes multas que puede ocasionar toda la mantis por el cumplimiento de los requerimientos del software.

Capítulo 3. Implementación

3.1. INTRODUCCIÓN A SELENIUM

3.1.1. Selenium 1.0

Selenium RC fue el principal proyecto de Selenium por un tiempo con el objetivo de promover librerías de funciones que permiten la automatización de pruebas de interfaz de usuario en aplicaciones web, antes de la unión con WebDriver/Selenium, que hoy en día es Selenium 2, podemos destacar que Selenium 1.0 ya es una versión no compatible. Los test creados en un lenguaje de programación hacen uso del API proporcionado por Selenium, donde simula la interacción de un cliente con la aplicación web a través de un navegador.

Selenium RC está dividido en dos componentes; el primero de ellos es Selenium Remote Control Server y Selenium Core.

3.1.2 Componentes RC

Los componentes RC están divididos en dos:

Según Selenium el servidor Selenium que inicia y elimina los navegadores, interpreta y ejecuta los comandos Selenese pasados desde el programa de prueba y actúa como un *proxy HTTP*, interceptando y verificando los mensajes HTTP pasados entre el navegador y el AUT.

Bibliotecas cliente proveen interfaz entre el lenguaje de programación y Selenium RC Server.

3.1.3. Selenium 2.0 (Selenium WebDriver)

Selenium 2.0 es una herramienta que facilita el desarrollo de pruebas automatizadas para aplicaciones web, es la fusión entre el proyecto de Selenium original y el proyecto WebDriver.

WebDriver aporta su API orientada a objetos para la interacción del Modelo de objetos de documento (DOM) y el control del navegador, así como sus diversas implementaciones de controladores de navegador.

Selenium tiene un buen soporte para los casos de uso más comunes, mientras que WebDriver ofrece múltiples posibilidades adicionales a través de su capacidad para salir del espacio aislado de JavaScript.

3.2. Introducción Implementación

En el presente proyecto se dará una introducción a la automatización de pruebas funcionales para entornos web, en este caso en el aplicativo de Pensiones con Selenium WebDriver bajo el IDE IntelliJ, usando como lenguaje de programación Java.

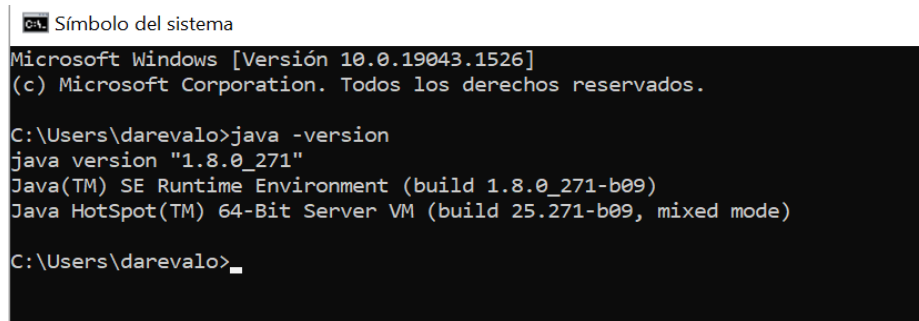
A continuación, se incluye una explicación de la configuración para hacer pruebas funcionales con Selenium y de las herramientas utilizadas, donde se muestra toda la parte de la instalación para entorno Windows, hasta la automatización de pruebas. Selenium es una Herramienta de automatización de pruebas de código abierto. El lenguaje con el que se trabajó para las pruebas automatizadas con Selenium es Java, por otro lado, Selenium puede soportar varios navegadores como FireFox, Chrome, Safari, el que se utilizó para las pruebas unitarias es el de Chrome.

Antes de iniciar con el proceso de creación de pruebas vamos a hacer la preparación del entorno, se dará una explicación de como instalar JAVA, descarga del IDE con el que trabajaremos que en este caso es IntelliJ, descarga de Maven para las dependencias, descarga del ChromeDriver y por último se describirá todo el proceso de pruebas.

3.3. Instalación Java

Instalación de la última versión disponible de JAVA

Antes de iniciar con la instalación se verifica si se tiene Java en el ordenador, para ello se abre un CMD donde se ejecuta el comando `java -version`, de la siguiente manera:



```
Símbolo del sistema
Microsoft Windows [Versión 10.0.19043.1526]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\darevalo>java -version
java version "1.8.0_271"
Java(TM) SE Runtime Environment (build 1.8.0_271-b09)
Java HotSpot(TM) 64-Bit Server VM (build 25.271-b09, mixed mode)

C:\Users\darevalo>
```

Ilustración 1CMD. Elaboración Propia.

en caso de no tenerlo vamos a iniciar con el proceso de instalación:

Lo primero que se hace es buscar en internet `java jdk download` y se abre el primer enlace de búsqueda. En la siguiente página de Oracle se puede realizar la descarga:

<https://www.oracle.com/java/technologies/downloads/#jdk18-windows>

Una vez que se descarga Java, se inicia con la configuración:

Se hace la búsqueda de las variables de entorno del sistema dentro del ordenador, una vez se ingresa, se muestra de la siguiente manera:

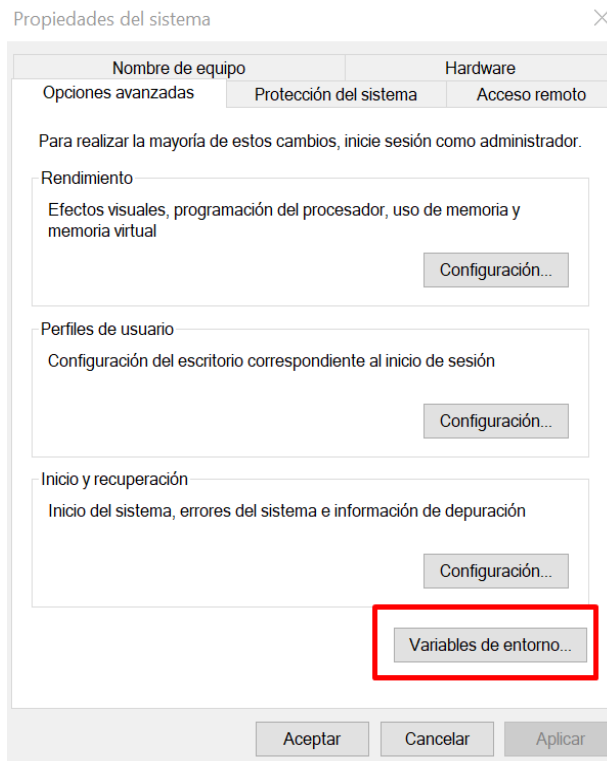


Ilustración 2. Variables de entorno. Elaboración Propia.

Se da click en variables de entorno configuramos Java Path con la ruta que se encuentra en
Program Files\Java\jdk1.8.0_281\bin

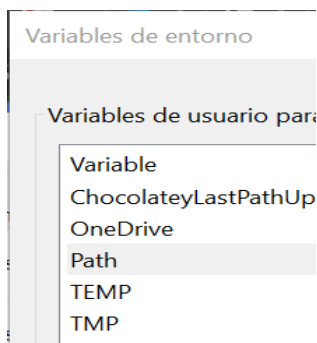


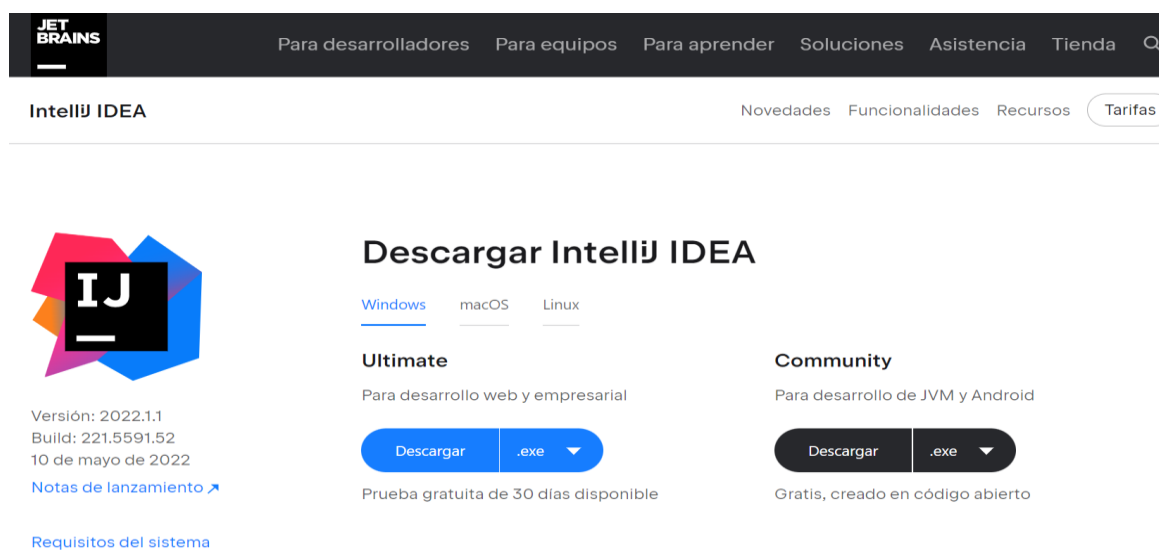
Ilustración 3. Variables de entorno. Elaboración Propia.

3.4. Instalación IntelliJ (IDE)

Instalación de la última versión disponible de IntelliJ

A continuación, se procede con la instalación del entorno de desarrollo (IDE) de IntelliJ, en el siguiente enlace se puede descargar:

<https://www.jetbrains.com/es-es/idea/download/#section=windows>



The screenshot shows the JetBrains website's download page for IntelliJ IDEA. The page features a dark navigation bar with the JetBrains logo and links for 'Para desarrolladores', 'Para equipos', 'Para aprender', 'Soluciones', 'Asistencia', and 'Tienda'. Below the navigation bar, the page title is 'IntelliJ IDEA' with links for 'Novedades', 'Funcionalidades', 'Recursos', and 'Tarifas'. The main content area is titled 'Descargar IntelliJ IDEA' and includes tabs for 'Windows', 'macOS', and 'Linux'. Under the 'Windows' tab, there are two options: 'Ultimate' (described as 'Para desarrollo web y empresarial') and 'Community' (described as 'Para desarrollo de JVM y Android'). Each option has a 'Descargar' button and a '.exe' dropdown menu. The 'Ultimate' option also includes a link to 'Prueba gratuita de 30 días disponible'. On the left side, there is a large IntelliJ logo and a section with version information: 'Versión: 2022.1.1', 'Build: 221.5591.52', and '10 de mayo de 2022', along with links for 'Notas de lanzamiento' and 'Requisitos del sistema'.

Ilustración 4: Descarga IntelliJ IDEA. Elaboración Propia.

Instalamos el IDE de IntelliJ y se siguen todos los pasos para su descarga exitosa.

3.5. Maven

Maven se puede definir como una herramienta de automatización de compilación y empaquetado, que se utiliza para administrar dependencia del proyecto que se esté realizando y todo el ciclo de vida del proyecto. Maven es creado por Apache Software Foundation y es utilizado en la construcción de software, Maven hace factible la creación de un software con dependencias incorporadas dentro del JAR. Es importante determinar las dependencias del

proyecto (Librerías externas) en un fichero, el POM (Project Object Model); este archivo viene en formato XML y contiene todo necesario cuando se genera el fichero ejecutable de la aplicación que se esté haciendo.

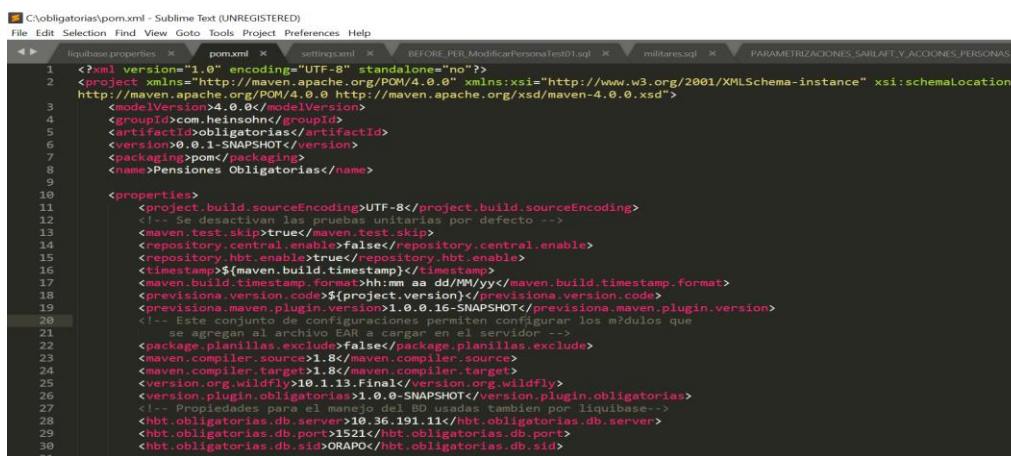
Para poder entender más acerca de Maven se describirán tres términos que serán importantes dentro de este proceso:

Repositorio local de Maven: Este repositorio local está en la maquina de la persona que este desarrollando, incluye el mecanismo del proyecto, al momento de ejecutar la compilación de Maven, los archivos JAR dependientes del proyecto se van a descargar en el repositorio local.

Repositorio central de Maven: es un repositorio predeterminado para Maven, esta consulta cuando alguna dependencia no está en el repositorio local y la descarga el repositorio central.

Modelo de objetos del proyecto: En POM XML se encontrara toda la configuración detallada que se va a utilizar para la compilación de Maven y se encuentra en la raíz del directorio del proyecto.

Ejemplo de pom.xml



```

1  <?xml version="1.0" encoding="UTF-8" standalone="no"?>
2  <project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="
3      http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
4      <modelVersion>4.0.0</modelVersion>
5      <groupId>com.heinsohn</groupId>
6      <artifactId>obligatorias</artifactId>
7      <version>0.0.1-SNAPSHOT</version>
8      <packaging>pom</packaging>
9      <name>Pensiones Obligatorias</name>
10
11     <properties>
12         <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
13         <!-- Se desactivan las pruebas unitarias por defecto -->
14         <maven.test.skip>true</maven.test.skip>
15         <repository.central.enable>false</repository.central.enable>
16         <repository.hbt.enable>true</repository.hbt.enable>
17         <timestamp>${maven.build.timestamp}</timestamp>
18         <maven.build.timestamp.format>hh:mm aa dd/MM/yy</maven.build.timestamp.format>
19         <previsiona.version.code>${project.version}</previsiona.version.code>
20         <previsiona.maven.plugin.version>1.0.0-SNAPSHOT</previsiona.maven.plugin.version>
21         <!-- Este conjunto de configuraciones permiten configurar los modulos que
22             se agregan al archivo EAR a cargar en el servidor -->
23         <package.planillas.exclude>false</package.planillas.exclude>
24         <maven.compiler.source>1.8</maven.compiler.source>
25         <maven.compiler.target>1.8</maven.compiler.target>
26         <version.org.wildfly>10.1.13.Final</version.org.wildfly>
27         <version.plugin.obligatorias>1.0.0-SNAPSHOT</version.plugin.obligatorias>
28         <!-- Propiedades para el manejo del BD usadas tambien por liquibase-->
29         <hbt.obligatorias.db.server>10.36.191.11</hbt.obligatorias.db.server>
30         <hbt.obligatorias.db.port>1521</hbt.obligatorias.db.port>
31         <hbt.obligatorias.db.sid>ORAPO</hbt.obligatorias.db.sid>

```

Ilustración 5: Pom.xml. Elaboración Propia.

Instalar Maven a través de línea de comandos

Una vez se instala y se configura las variables de entorno de Java, se procede con la descarga Maven y configuración de las variables de entorno de Maven. Se puede descargar Maven en el siguiente enlace:

<https://maven.apache.org/download.cgi>


| | | | |
|---|--|--|---|
| Disk | Approximately 10MB is required for the Maven installation itself. In addition to that, additional disk space will be used for your local Maven repository. The size of your local repository will vary depending on usage but expect at least 500MB. | | |
| Operating System | No minimum requirement. Start up scripts are included as shell scripts and Windows batch files. | | |
| Files | | | |
| Maven is distributed in several formats for your convenience. Simply pick a ready-made binary distribution archive and follow the installation instructions . Use a source archive if you intend to build Maven yourself. | | | |
| In order to guard against corrupted downloads/installations, it is highly recommended to verify the signature of the release bundles against the public KEYS used by the Apache Maven developers. | | | |
| | Link | Checksums | Signature |
| Binary tar.gz archive | apache-maven-3.8.5-bin.tar.gz | apache-maven-3.8.5-bin.tar.gz.sha512 | apache-maven-3.8.5-bin.tar.gz.asc |
| Binary zip archive |  apache-maven-3.8.5-bin.zip | apache-maven-3.8.5-bin.zip.sha512 | apache-maven-3.8.5-bin.zip.asc |
| Source tar.gz archive | apache-maven-3.8.5-src.tar.gz | apache-maven-3.8.5-src.tar.gz.sha512 | apache-maven-3.8.5-src.tar.gz.asc |
| Source zip archive | apache-maven-3.8.5-src.zip | apache-maven-3.8.5-src.zip.sha512 | apache-maven-3.8.5-src.zip.asc |

Ilustración 6: Descarga Maven. Elaboración Propia.

Una vez que se descarga Maven, se procede con la configuración de las variables de entorno:

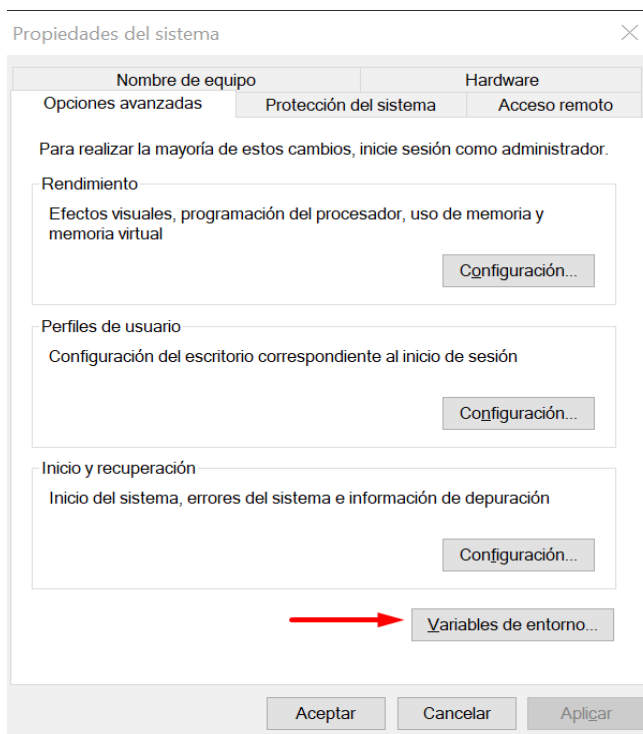


Ilustración 7: Variables de entorno. Elaboración Propia.

Se ingresa a variables de entorno, se da click en “Nueva” y se ingresa el nombre de la variable de entorno y la ruta donde se guardó, se da click en aceptar y se guarda la variable de entorno.

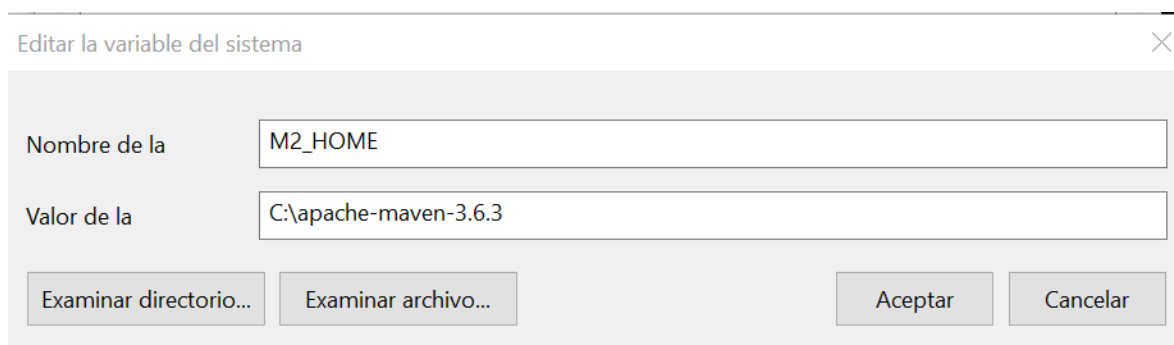


Ilustración 8: Editar Variable Sistema. Elaboración Propia.

Por último, si Maven se instaló con éxito en el ordenador se puede verificar que quedo instalado correctamente, ingresando a Símbolo del sistema y se ejecuta la siguiente línea de comando `mvn -version`, debe salir de la siguiente manera:

```
C:\Users\darevalo>mvn -version
Apache Maven 3.6.3 (cecedd343002696d0abb50b32b541b8a6ba2883f)
Maven home: C:\apache-maven-3.6.3\bin\..
Java version: 1.8.0_271, vendor: Oracle Corporation, runtime: C:\Java\jdk1.8.0_271\jre
Default locale: es_CO, platform encoding: Cp1252
OS name: "windows 10", version: "10.0", arch: "amd64", family: "windows"
```

Ilustración 9: Símbolo del Sistema. Elaboración Propia.

También se puede verificar la instalación en el IDE de IntelliJ ingresando a File >> Settings

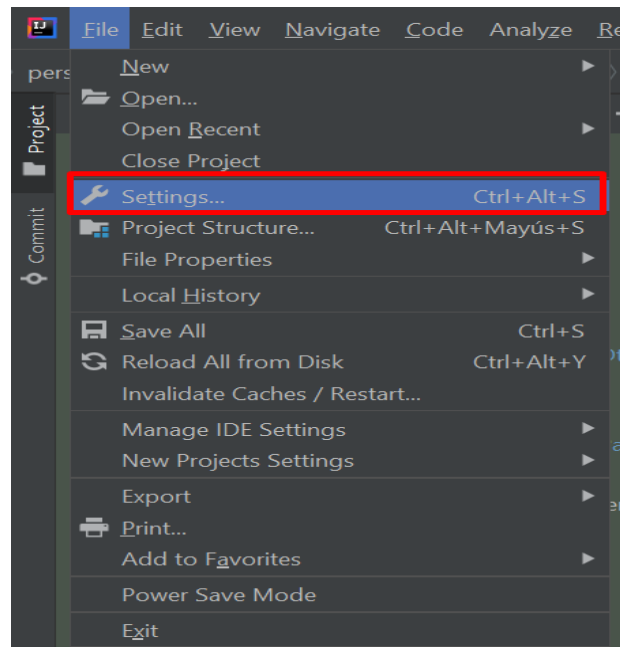


Ilustración 10: Settings. Elaboración Propia.

Debería salir una pantalla como la siguiente:

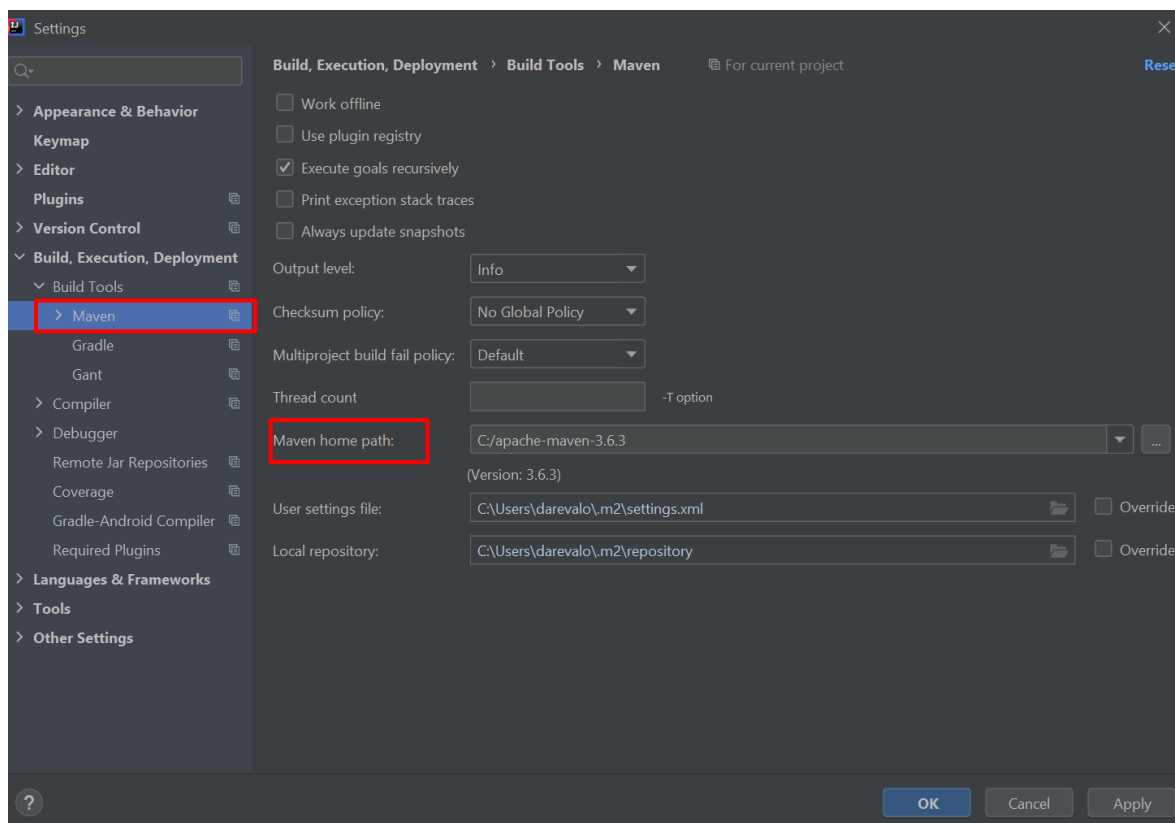


Ilustración 11. Ajustes Maven IntelliJ. Elaboración Propia.

Las dependencias que se deben tener en el POM son las siguientes

```
<dependency>
  <groupId>org.testng</groupId>
  <artifactId>testng</artifactId>
  <version>7.0.0</version>
  <scope>test</scope>
</dependency>
<dependency>
  <groupId>org.apache.maven</groupId>
  <artifactId>maven-model</artifactId>
  <version>3.5.3</version>
</dependency>
```

Ilustración 12. Dependencias POM. Elaboración Propia.

Ya terminada la verificación se puede continuar con el proceso de creación del proyecto.

Para el inicio de las pruebas se importó el proyecto de Selenium del módulo de personas en el cual se estará trabajando, se presenta de la siguiente forma:

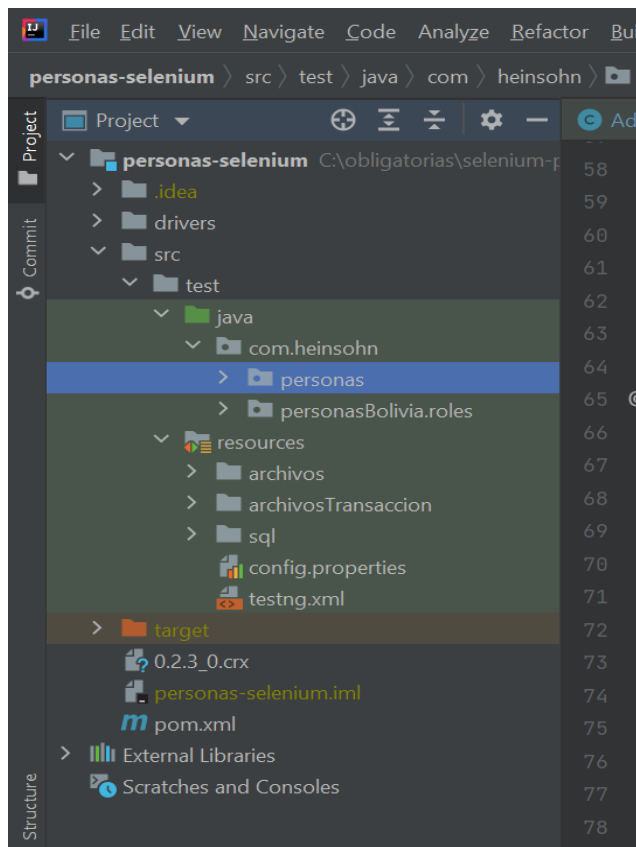


Ilustración 13. Carpetas Proyectos. Elaboración Propia.

3.6. ChromeDriver

Antes de iniciar con la descarga de Chrome driver lo primero que se hace es verificar que versión de navegador se tiene, para ello en la parte derecha del navegador se despliega el menú y entramos a la opción de configuración, como se muestra a continuación:

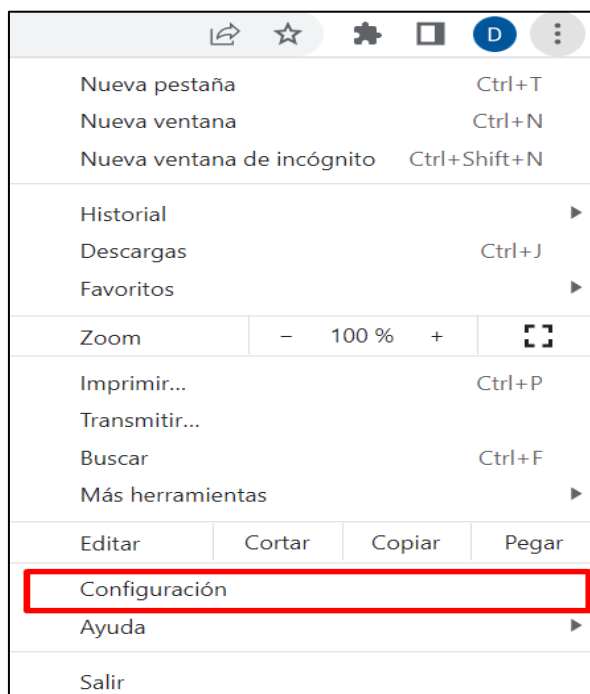


Ilustración 14. Configuración ChromeDriver. Elaboración Propia.

Una vez que se ingresa a la parte de la configuración, se encontrara la opción de “Acerca de Chrome” se da click en esa sección:

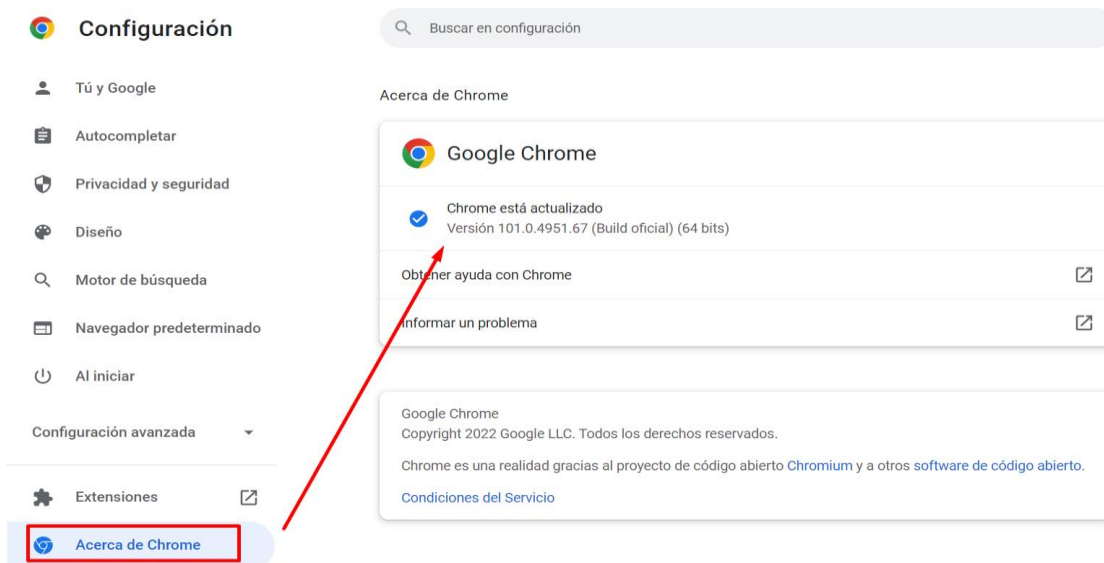


Ilustración 15. Versión Chrome. Elaboración Propia.

Como se puede evidenciar en la ilustración 15 en esa sección se puede ver la versión que se tiene, una vez que se sabe la versión del navegador, se procede a descargar el Chrome Driver que más se acerque a la versión que tiene el navegador.

Para descargar Chrome Driver se busca en la web “Chrome Driver” y se ingresa al primer link de búsqueda que sale que en este caso sería el siguiente:

<https://chromedriver.chromium.org/downloads>

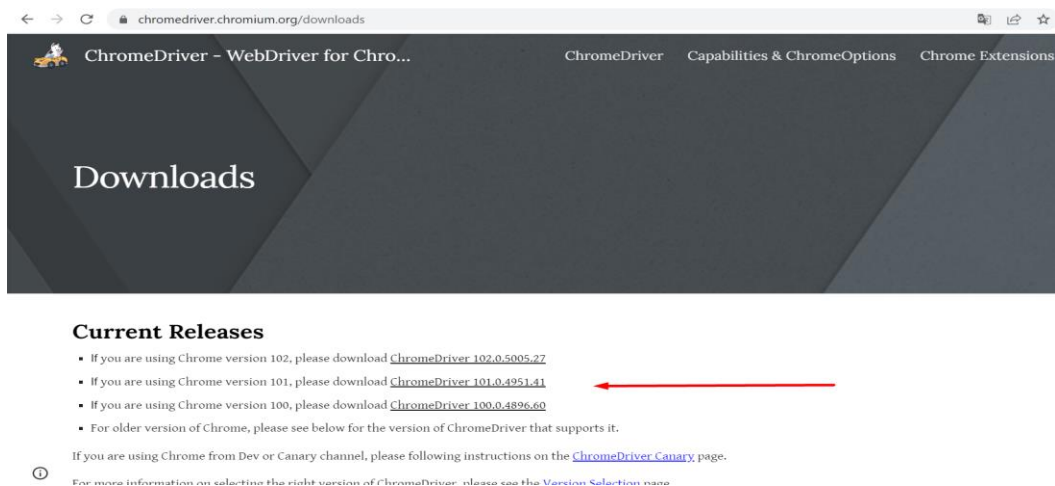


Ilustración 16. Chrome Driver. Elaboración Propia.

Se da click en la versión que se tiene del navegador o la que más se acerque y se abre otra ventana donde se podrá elegir el ChromeDriver que se quiere descargar:

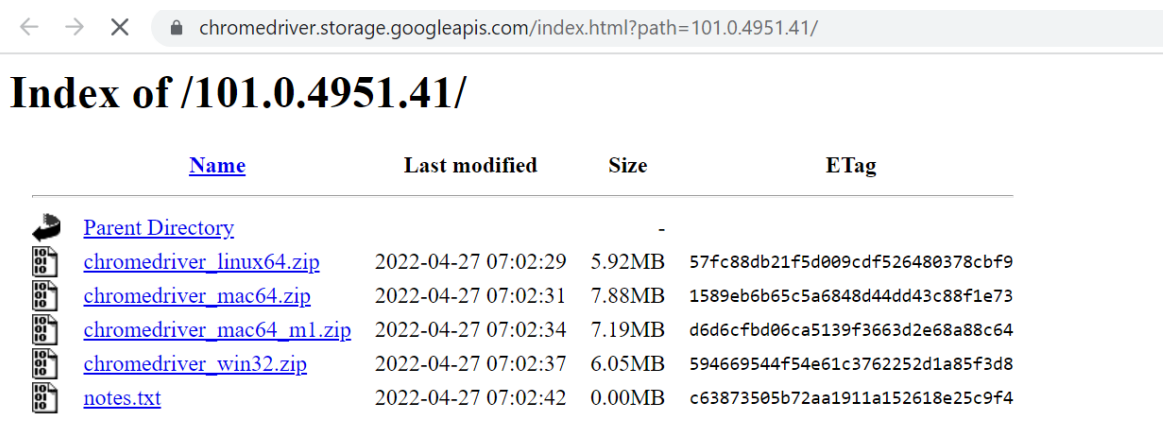
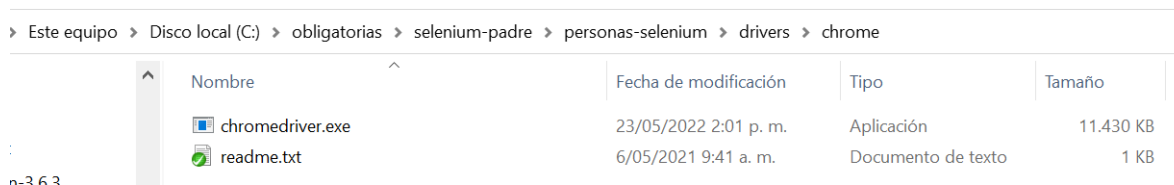


Ilustración 17. Versiones Chrome Driver. Elaboración Propia.

Se descarga el Chrome Driver compatible con el sistema operativo, una vez descargado se descomprime el archivo y se agrega el chromedriver.exe en la carpeta donde se tiene los drivers.



| Nombre | Fecha de modificación | Tipo | Tamaño |
|------------------|-----------------------|--------------------|-----------|
| chromedriver.exe | 23/05/2022 2:01 p. m. | Aplicación | 11.430 KB |
| readme.txt | 6/05/2021 9:41 a. m. | Documento de texto | 1 KB |

Ilustración 18. Carpeta drivers. Elaboración Propia.

Después de que se tiene el driver, se procede con la creación de la carpeta con el nombre de las pruebas que se van a realizar.

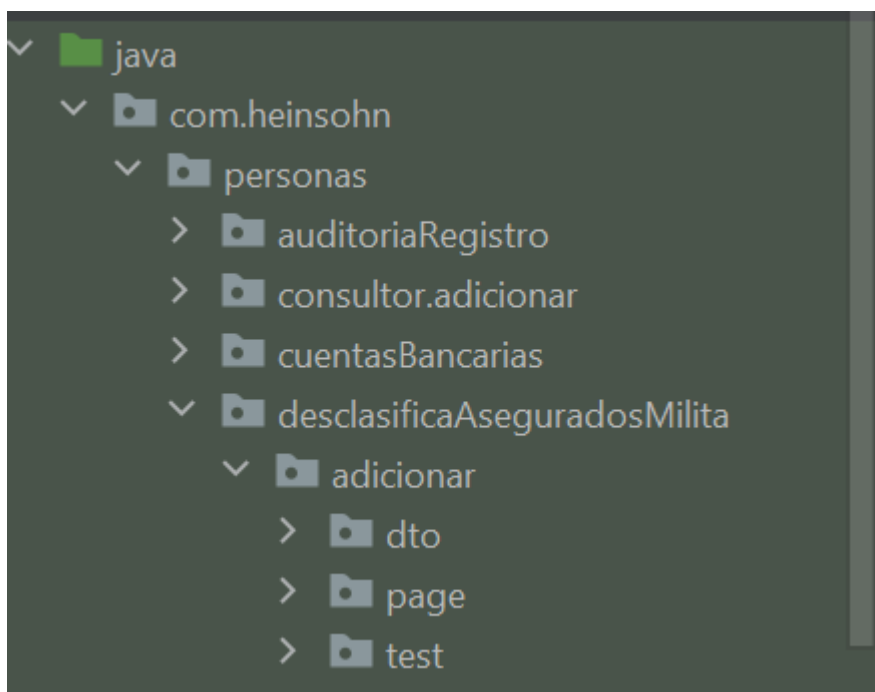


Ilustración 19. Carpeta Proyecto. Elaboración Propia.

Antes de explicar cómo se distribuyen las clases de pruebas que se crearon, se dará una pequeña introducción a unos conceptos que estaremos utilizando a lo largo de este proyecto que son los métodos GET y SET.

Java es un lenguaje de programación orientado a objetos, el código orientado a objetos se construye mediante las clases, que modelan la creación de un objeto. Un objeto es un conjunto de variables que pueden ser privadas y solo se pueden acceder a ellas por el objeto que tienen y métodos que tienen un comportamiento, los métodos de get y set son métodos que interactúan con las variables privadas.

El método get de las clases se utiliza para mostrar y el método set para modificar, se puede generar automáticamente estos métodos, dando click encima de la variable, saldrá una ventana donde se dará click en Generate:

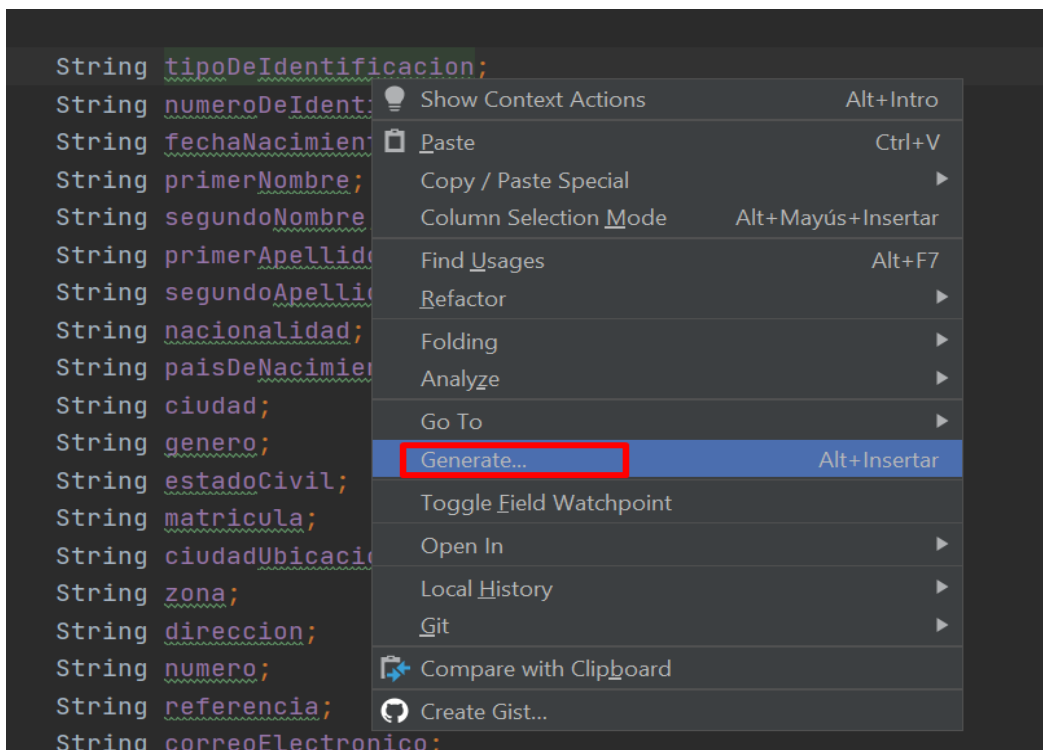


Ilustración 20. Generación automática métodos GET y SET. Elaboración Propia.

Posteriormente se le da click en Generatem, se elige Getter and Setter

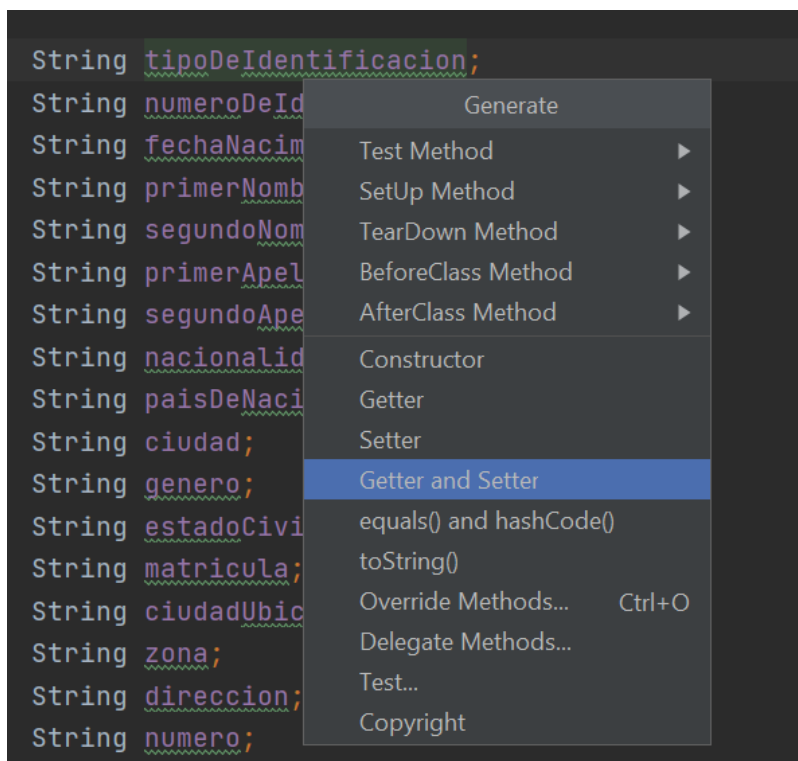


Ilustración 21. Métodos Getter and Setter. Elaboración Propia.

Se generará automáticamente los métodos de GET y SET del atributo del objeto que se tiene

```
public String getTipoDeIdentificacion() {
    return tipoDeIdentificacion;
}

public void setTipoDeIdentificacion(String tipoDeIdentificacion) {
    this.tipoDeIdentificacion = tipoDeIdentificacion;
}
```

Ilustración 22. Métodos GET y SET. Elaboración Propia.

Estos métodos son utilizados en la programación orientada a objetos, el método set como podemos evidenciar en la ilustración 22 es un método público, en cual se ocupa de darle un valor a un atributo de un objeto, que esta encapsulado en la clase que tenemos. Por otro lado, el

método get es un método público que también que se encarga de mostrar un atributo de un objeto, el cual estará encapsulado en la clase.

Después de esta pequeña introducción acerca de los métodos GET y SET se continua con la creación de las clases, como evidencia en la ilustración 23:

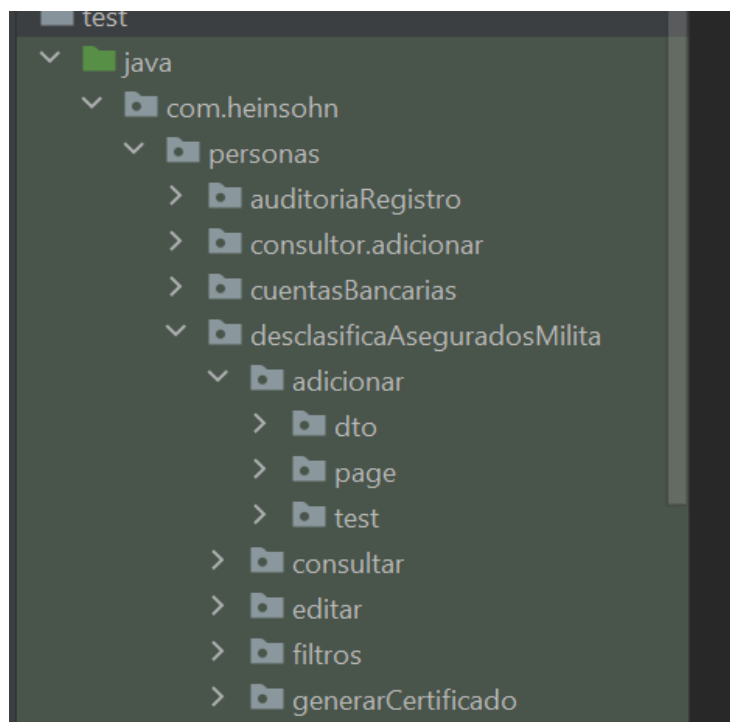


Ilustración 23. Clases Proyecto. Elaboración Propia.

Se creo una carpeta llamada desclasificaAseguradosMilita que es una de las funcionalidades de las cuales se hicieron pruebas automatizadas con Selenium, esta carpeta está dividida en tres partes; las clases dto, pages y test que las definiremos a continuación:

3.7. Data Transfer Object (DTO)

El acceso de datos que se hace en a través de Data Acces Object (DAO), con esto tenemos la abstracción en modelo de datos donde solo podemos acceder ellos a través de métodos definidos. Los Data Acces Object introducen los DTO, los Data transfer Object sirven para llevar datos, son un tipo de objeto y contienen propiedades del objeto. Todos estos datos unen a una instancia de un JavaBean. Los dto son objetos que determinan como se enviaran los datos en la red y se conforman de una serie de atributos. Los objetos de un DTO son de solo lectura, se utilizan para transferir entre el cliente y el servidor, en esta clase dto es importante evitar que se hagan métodos que realicen cálculos sobre los datos que tenemos, solo tendremos en estas clases los métodos de GET y SET de los atributos que tengamos en el DTO.

A continuación, se crea un dto con los atributos que contiene una pantalla, si se cambia de pantalla se debe crear un nuevo dto con todos los atributos de la nueva pantalla, se mostrara un ejemplo de una clase dto como vemos en la ilustración 24:

```
5 public class AdicionarDatosAseguradoDto {
6
7     String tipoDeIdentificacion;
8     String numeroDeIdentificacion;
9     String fechaNacimiento;
10    String primerNombre;
11    String segundoNombre;
12    String primerApellido;
13    String segundoApellido;
14    String nacionalidad;
15    String paisDeNacimiento;
16    String ciudad;
17    String genero;
18    String estadoCivil;
19    String matricula;
20    String ciudadUbicacion;
21    String zona;
22    String direccion;
23    String numero;
24    String referencia;
25    String correoElectronico;
26    String tipoDeCorreoElectronico;
27    String celular;
28    String telefono;
29    String extension;
```

Ilustración 24. Clase DTO. Elaboración Propia.

```
31     /**
32      * @return the tipoDeIdentificacion
33      */
34     public String getTipoDeIdentificacion() { return tipoDeIdentificacion; }
37
38     /**
39      * @param tipoDeIdentificacion the tipoDeIdentificacion to set
40      */
41     public void setTipoDeIdentificacion(String tipoDeIdentificacion) {
42         this.tipoDeIdentificacion = tipoDeIdentificacion;
43     }
44
45     /**
46      * @return the numeroDeIdentificacion
47      */
48     public String getNumeroDeIdentificacion() { return numeroDeIdentificacion; }
51
52     /**
53      * @param numeroDeIdentificacion the numeroDeIdentificacion to set
54      */
55     public void setNumeroDeIdentificacion(String numeroDeIdentificacion) {
56         this.numeroDeIdentificacion = numeroDeIdentificacion;
57     }
```

Ilustración 25. Clase DTO. Elaboración Propia.

En esta clase dto podemos se puede evidenciar los atributos con sus respectivos métodos get y set.

3.8. Page Object Model (POM)

Page Object encapsula todo el comportamiento de las páginas del aplicativo web, esto nos facilita poder realizar tests, el uso de los pages son muy importantes a la hora de realizar pruebas funcionales automatizadas, ya que dentro de estas clases se puede modelar el comportamiento de estas páginas a través de diferentes métodos que se pueden tener, para manipular elementos de la página, como ingresar a una determinada función, cabe resaltar que si algunas funcionalidades que tiene la página web sufre algún tipo de cambio, se tendrá que cambiar los métodos en los pages que sufrieron cambios.

Para dar mayor claridad a lo descrito anteriormente se dará un ejemplo de cómo podría llegar a ser un page:

```

11  /**
12  * <b> Descripción:<b> Clase que contiene acciones y elementos de la pantalla de datos de asegurado
13  * del rol tercero <b> Caso de uso:<b> CUPE1435 OBL Crear-Editar solicitud de desclasificación de militar
14  *
15  * @author Danna Michelle Arévalo Hernández <darevalo@heinsohn.com.co>
16  * @version
17  *
18  */
19  public class AdicionarDatosAseguradoPage {
20
21  // Aca ingresamos los atributos de la pantalla con sus localizadores que en este caso es el id
22  // de cada elemento de la pantalla
23  private By tipoDeIdentificacion = By.id("idTipoIdentificacion");
24  private By numeroDeIdentificacion = By.id("idNumeroIdentificacion");
25  private By fechaNacimiento = By.id("fechaNacimientoInput");
26  private By primerNombre = By.id("primerNombreId");
27  private By segundoNombre = By.id("txtSegundoNombreId");
28  private By primerApellido = By.id("txtPrimerApellidoId");
29  private By segundoApellido = By.id("txtSegundoApellidoId");
30  private By nacionalidad = By.id("idNacionalidad");
31  private By paisDeNacimiento = By.id("idPais");
32  private By ciudad = By.id("idCiudadNacimiento");
33  private By genero = By.id("idGenero");
34  private By estadoCivil = By.id("idEstadoCivil");
35  private By matricula = By.id("matriculaId");
36  private By acordeon = By.id("mdb-accordion-item-head-empleador");

```

Ilustración 26. Clase Page. Elaboración Propia.

```

37  private By ciudadUbicacion = By.id("filtroAutoCompletarObjetoLista");
38  private By zona = By.id("zona");
39  private By direccion = By.id("direccion");
40  private By numero = By.id("numero");
41  private By referencia = By.id("referencia");
42  private By correoElectronico = By.id("correoElectronico");
43  private By tipoDeCorreoElectronico = By.id("selectTipoCorreo");
44  private By celular = By.id("celular");
45  private By telefono = By.id("telefono");
46  private By extension = By.id("extension");
47  private By btnSiguiente = By.id("wzSiguienteID");
48
49
50  //Declaramos los objetos WebDriver
51  private final WebDriver driver;
52  private final SeleniumWebDriverUtils utils;
53
54  //Constructor del page donde le pasamos el objeto WebDriver
55  public AdicionarDatosAseguradoPage(WebDriver driver) {
56      this.driver = driver;
57      utils = new SeleniumWebDriverUtils(this.driver);
58  }
59

```

Ilustración 27. Clase Page. Elaboración Propia.

```

58  /**
59      *
60      * Metodo encargado de ingresar todos los datos de la solicitud, vamos a pasarle al metodo el dto
61      *
62      * @author Danna Michelle Arevalo Hernandez <darevalo@heinsohn.com.co>
63      *
64      */
65  public void datosAsegurado (AdicionarDatosAseguradoDto adicionarDatosAseguradoDto) {
66
67      // Tiempo de espera pre determinado
68      utils.waitForLoading(Integer.parseInt(SeleniumProperties.getPropiedad(ConfigSeleniumPropiedades.TIME));
69
70      // Ingresamos el tipo de identificacion
71      utils.selectOptionList(tipoDeIdentificacion, adicionarDatosAseguradoDto.getTipoDeIdentificacion());
72      utils.waitForOneSeconds();
73      utils.waitForOneSeconds();
74
75      // Ingresamos el numero de identificacion
76      utils.type(numeroDeIdentificacion, adicionarDatosAseguradoDto.getNumeroDeIdentificacion());
77      utils.waitForOneSeconds();
78      utils.waitForOneSeconds();
79      utils.waitForOneSeconds();
80

```

Ilustración 28. Método Clase Page. Elaboración Propia.

Lo primero que se definió dentro de la clase page son los atributos o las variables que tiene una pantalla dentro del aplicativo, donde se podrá buscar estos atributos a través del localizador By, en este ejemplo utilizamos el id que tenga cada elemento en la página web, este id debe ser único para cada elemento; así de esta forma permitirá que la prueba encuentre de manera más fácil cada elemento que esté dentro de la pantalla y que se vaya a utilizar en los métodos del page, más adelante se dará a conocer todo acerca de los localizadores.

En segunda instancia se hace el constructor de la clase, declaramos un objeto WebDriver y se los pasamos en el constructor, por último, dentro de estos Pages tendremos los métodos que tendrán las acciones que hacen cada elemento dentro de la pantalla para poder realizar todo el proceso de prueba de cómo el usuario interactuaría con la aplicación.

3.9. TestNG

TestNG son pruebas unitarias de código abierto, nos permite la ejecución de casos de pruebas automatizados demasiados complejos y nos brindan diversas funcionalidades. Este se fundamenta en ideas de diseño de Junit y NUnit, nos brinda funcionalidades más sencillas y eficaces. Una de las ventajas que nos ofrece TestNG es la posibilidad de ejecución de scripts dentro de las pruebas automatizadas, también nos permite la reutilización de casos de prueba para diversos grupos de datos, puede soportar varias pruebas múltiples y genera informes de las pruebas automatizadas. Debemos agregar al archivo pom.xml la dependencia de TestNG:

```
<dependency>  
  <groupId>org.testng</groupId>  
  <artifactId>testng</artifactId>  
  <version>7.0.0</version>  
  <scope>test</scope>  
</dependency>
```

Ilustración 29. Dependencias. Elaboración Propia.

Las anotaciones del TestNG de los casos de prueba se dividen en Test Suite con una o más pruebas, las pruebas que son de una o más clases de prueba y la última las clases que se componen de uno o más métodos de prueba, la estructura que tiene que ir en el archivo testing.xml es la siguiente:

```

401 <test name="SolicitudDeclaracionAseguradoMilitar" parallel="tests" thread-count="20" preserve-order="true">
402   <groups>
403     <run>
404       <include name="solicitudDesclasificacionAsegurado"/>
405     </run>
406   </groups>
407
408   <classes>
409     <class name="com.heinsohn.personas.desclasificaAseguradosMilita.adicionar.test.AdicionarSolicitudDesclaAseguradosTest01"/>
410     <class name="com.heinsohn.personas.desclasificaAseguradosMilita.consultar.ConsultarDesclaAseguradosTest01"/>
411     <class name="com.heinsohn.personas.desclasificaAseguradosMilita.filtros.test.FiltrarAseguradoTest01"/>
412     <class name="com.heinsohn.personas.desclasificaAseguradosMilita.editar.test.EditarSolicitudDesclaAseguradosTest01"/>
413
414     <class name="com.heinsohn.personas.desclasificaAseguradosMilita.generarCertificado.GenerarCertificacionTest01"/>
415
416   </classes>
417 </test>
418
419 </suite>

```

Ilustración 30. TestNG. Elaboración Propia.

Cuando se ejecutan las pruebas automáticas se puede utilizar diversas anotaciones como la inicialización antes de la prueba, ejecución de caso de prueba que se tenga y por el ultimo la limpieza posterior a la ejecución de la prueba.

3.10. Anotaciones Comunes

| Nombre de la anotación | Significado de la anotación |
|------------------------|--|
| @BeforeSuite | El método anotado solo se ejecutará una vez, antes de ejecutar todas las pruebas en este conjunto. |
| @AfterSuite | El método anotado solo se ejecutará una vez después de que se hayan ejecutado todas las pruebas en este conjunto. |
| @BeforeClass | El método anotado solo se ejecutará una vez antes de la llamada al método en la clase actual. |
| @AfterClass | El método anotado solo ejecutará todos los métodos de prueba en la clase actual después de ejecutarlo una vez. |
| @BeforeTest | El método anotado se ejecutará antes de que cualquier método de prueba pertenezca a la ejecución de la etiqueta <test> de la clase interna. |
| @AfterTest | Después de ejecutar el método anotado, todos los métodos de prueba pertenecen a la clase interna de la etiqueta <test>. |
| @BeforeGroups | Lista de grupos, este método de configuración se ejecutará antes. Se garantiza que se llamará a este método cuando se ejecute el primer método de prueba que pertenezca a cualquiera de estos grupos. |
| @AfterGroups | La lista de grupos se ejecutará después de este método de configuración. Se garantiza que este método se ejecutará poco después de que se llame al último método de prueba que pertenece a cualquiera de estos grupos. |
| @BeforeMethod | Los métodos anotados se ejecutarán antes de cada método de prueba. |
| @AfterMethod | El método anotado se ejecutará después de cada método de prueba. |
| @DataProvider | Marque un método, un método de prueba para proporcionar datos. El método anotado debe devolver un Objeto [], donde cada objeto [] puede asignarse en la lista de parámetros del método de prueba. El método @Test, con la esperanza de recibir datos de este DataProvider, necesita usar un nombre de dataProvider igual al nombre de esta anotación. |

| | |
|--------------------|--|
| @Factory | Como fábrica, el objeto de la clase de prueba que devuelve TestNG se usará para marcar el método. El método debe devolver Object []. |
| @Listeners | Definir un oyente para la clase de prueba. |
| @Parameters | Describe cómo pasar parámetros al método @Test. |
| @Test | Marcar una clase o método como parte de la prueba. |

Ilustración 31. Nota. Adaptado de Programmerclick.com. 2022. Notas de estudio de selenio-marco de prueba de unidad-TestNG - programador clic. Disponible en: <<https://programmerclick.com/article/2681954615/>>

3.11. Implementación Test

```

26 /**
27  * Clase encargada de adicionar una solicitud de desclasificación de asegurados militares
28  * <b>Caso de Uso</b> CUPE0234
29  *
30  * @author Danna Michelle Arévalo Hernández <darevalo@heinsohn.com.co>
31  */
32
33 public class AdicionarSolicitudDesclaAseguradosTest01 {
34
35     private SeleniumTest selenium;
36     private LoginPage loginPage;
37     private MenuPage menuPage;
38     private AccionesDesclaAseguradosPage accionesDesclaAseguradosPage;
39     private AdicionarSolicitudPage adicionarSolicitudPage;
40     private AdicionarDatosAseguradoPage adicionarDatosAseguradoPage;
41     private EntrarCardPersonasUtils entrarCardPersonasUtils;
42
43     List<String> filaEncontrada;
44

```

Ilustración 32. Clase Test. Elaboración Propia.

```

45
46 @BeforeMethod(groups = { "solicitudDesclasificacionAsegurado" })
47 public void iniciarPrueba() {
48     // Se evalua que el objeto selenium no tenga instancias previas
49     if (selenium == null) {
50         selenium = new SeleniumTest(getClass().getSimpleName());
51         loginPage = new LoginPage(selenium.getDriver());
52         menuPage = new MenuPage(selenium.getDriver());
53         accionesDesclaAseguradosPage = new AccionesDesclaAseguradosPage(selenium.getDriver());
54         adicionarSolicitudPage = new AdicionarSolicitudPage(selenium.getDriver());
55         adicionarDatosAseguradoPage = new AdicionarDatosAseguradoPage(selenium.getDriver());
56         entrarCardPersonasUtils = new EntrarCardPersonasUtils(selenium.getDriver());
57         filaEncontrada = new ArrayList<>();
58     }
59
60 }
61
62 }

```

Ilustración 33. Clase Test. Elaboración Propia.

```

68      /**
69       * Metodo encargado de crear una solicitud de desclasificacion de asegurados militares
70       *
71       * @author Danna Michelle Arevalo Hernandez <darevalo@heinsohn.com.co>
72       *
73       */
74      @Test(groups = { "solicitudDesclasificacionAsegurado" })
75      public void adicionarSolictudDescLaAseguradosTest01() {
76
77          // Ingresar a Personas
78          loginPage.login(SeleniumProperties.getPropiedad(ConfigSeleniumPropiedades.USUARIO),
79                          SeleniumProperties.getPropiedad(ConfigSeleniumPropiedades.CONTRASENA));
80
81          selenium.getUtils().waitForLoading(Integer.parseInt(SeleniumProperties.getPropiedad(ConfigSeleniumPropiedades.WAIT_FOR_LOADING_PERSONAS)));
82          selenium.getUtils().waitForLoading(Integer.parseInt(SeleniumProperties.getPropiedad(ConfigSeleniumPropiedades.WAIT_FOR_LOADING_PERSONAS)));
83
84          selenium.getUtils().waitForLoading(Integer.parseInt(SeleniumProperties.getPropiedad(ConfigSeleniumPropiedades.WAIT_FOR_LOADING_PERSONAS)));
85          selenium.getUtils().waitForLoading(Integer.parseInt(SeleniumProperties.getPropiedad(ConfigSeleniumPropiedades.WAIT_FOR_LOADING_PERSONAS)));
86
87          menuPage.clickOnMenu(menuPage.personas, menuPage.gestionar);
88
89          // En el modulo de gestionar entramos a la card de personas
90          entrarCardPersonasUtils.entrarCardAsegurados();
91

```

Ilustración 34. Método Clase Test. Elaboración Propia.

```

92          // Le damos click en la card de desclasificacion de asegurados
93          accionesDescLaAseguradosPage.entrarCardDeclasiAsegurados();
94
95          // Le damos click en el boton de adicionar
96          accionesDescLaAseguradosPage.clickBtnAdicionar();
97
98          // Ingresamos los datos de la solicitud en el primer stiper
99          AdicionarSolictudDto adicionarSolictudDto = new AdicionarSolictudDto();
100         adicionarSolictudDto.setCiteMinisterio("91");
101         adicionarSolictudDto.setFechaDeCite("28/02/2022");
102         adicionarSolictudDto.setFechaDeRecepcion("22/02/2022");
103         adicionarSolictudDto.setCua("567");
104         adicionarSolictudDto.setGestor("312-MARISOL BAUTISTA");
105         adicionarSolicitudPage.datosSolicitud(adicionarSolictudDto);
106
107         // Ingresamos los datos de datos de asegurado en el segundo stiper
108         AdicionarDatosAseguradoDto adicionarDatosAseguradoDto = new AdicionarDatosAseguradoDto();
109         adicionarDatosAseguradoDto.setTipoDeIdentificacion("Carnet de extranjero (E)");
110         adicionarDatosAseguradoDto.setNumeroDeIdentificacion("4000500");
111         adicionarDatosAseguradoDto.setFechaNacimiento("20/01/1969");
112         adicionarDatosAseguradoDto.setPrimerNombre("RODRIGO");
113         adicionarDatosAseguradoDto.setSegundoNombre("ANDRES");
114         adicionarDatosAseguradoDto.setPrimerApellido("GOMEZ");
115         adicionarDatosAseguradoDto.setSegundoApellido("AMEZQUITA");
116         adicionarDatosAseguradoDto.setNacionalidad("BOLIVIA");

```

Ilustración 35. Método Clase Test. Elaboración Propia.

```

117     adicionarDatosAseguradoDto.setPaisDeNacimiento("BOLIVIA");
118     adicionarDatosAseguradoDto.setCiudad("SUCRE / DROPEZA / CHUQUISACA");
119     adicionarDatosAseguradoDto.setGenero("Masculino");
120     adicionarDatosAseguradoDto.setEstadoCivil("SOLTERO(A)");
121     adicionarDatosAseguradoDto.setMatricula("6901206MR");
122     adicionarDatosAseguradoDto.setCiudadUbicacion("COCHABAMBA / CERCADO / COCHABAMBA");
123     adicionarDatosAseguradoDto.setZona("Queru queru");
124     adicionarDatosAseguradoDto.setDireccion("Av. Americas");
125     adicionarDatosAseguradoDto.setNumero("500");
126     adicionarDatosAseguradoDto.setReferencia("Pruebas");
127     adicionarDatosAseguradoDto.setCorreoElectronico("pruebas@gestora.com.bo");
128     adicionarDatosAseguradoDto.setTipoDeCorreoElectronico("Personal");
129     adicionarDatosAseguradoDto.setCelular("3202272833");
130     adicionarDatosAseguradoPage.datosAsegurado(adicionarDatosAseguradoDto);
131
132     //Cerramos sesion
133     LoginPage.logout();
134 }
135

```

Ilustración 36. Método Clase Test. Elaboración Propia.

```

132 /**
133  *
134  * Metodo encargado de cerrar la prueba
135  *
136  * @author Danna Michelle Arevalo Hernandez <darevalo@heinsohn.com.co>
137  *
138  */
139 @AfterMethod(groups = { "solicitudDescLasificacionAsegurado" })
140 public void cerrar(){
141     //Cerramos el navegador
142     selenium.getDriver().close();
143
144     // Script para borrar los datos ingresados en la prueba
145     selenium.getUtilsDB().executeSQLFile(
146         fileName: "src/test/resources/sql/personas/descLasificacionAsegurados/AFTER_PER_AdicionarSolicitudDescLaAseguradosTest01.sql",
147         ConstantesMicroservicios.PERSONAS);
148 }
149 }
150
151

```

Ilustración 37. AfterMethod Clase Test. Elaboración Propia.

En la primera parte de la clase test se tiene los atributos privados que extienden de otras clases donde existen los métodos que diligencian el login con username y password, también el ingreso al menú donde está el módulo de personas, después se encuentra la anotación @BeforeMethod que nos sirve para ejecutar el método que tenga antes del método de prueba, dentro de método del @BeforeMethod se inicializa los atributos y se evalúa que el objeto Selenium no tenga instancias previas.

Posteriormente se tiene la anotación `@Test` para ponerlo como método de la clase, acá estará el método principal de la clase donde se encuentran las acciones que se van a realizar en el aplicativo, dentro del método se ingresa al módulo de personas, se diligencia la parte del login, se ingresa al menú, hace el llamado a los métodos que se tienen en el page que realizan alguna acción en específico, si se tiene que ingresar algún tipo de dato se utiliza el método `SET`.

Para culminar la prueba se encuentra la anotación `@AfterMethod`, que es el método que se ejecutara después de todos los métodos de la prueba, en esta parte se cierra el navegador y se ejecuta el script para borrar los datos que se crearon dentro de la prueba, para esta parte se debe crear una carpeta aparte donde tendrá todos los archivos sql con los datos que se pueden ejecutar antes del método principal de la prueba o después de él, así de esta forma se llama la ruta donde se encuentra el script de ejecución.

3.12. Interfaz WebDriver (Localización de Elementos Web)

Todos los elementos de una página web como botones, links, imágenes, cajas de textos se llama `WebElements` para Selenium `WebDriver`.

Dentro de la parte de `Front-End` del aplicativo se puede localizar todos estos elementos `Web`, para ello se utiliza el método de `driver.findElement()` que devuelve el `WebElement` localizado a través de los localizadores `By` que los veremos más adelante. Este método permite localizar un elemento en la página web y realizar diferentes acciones sobre él.

Se puede hacer de la siguiente forma:

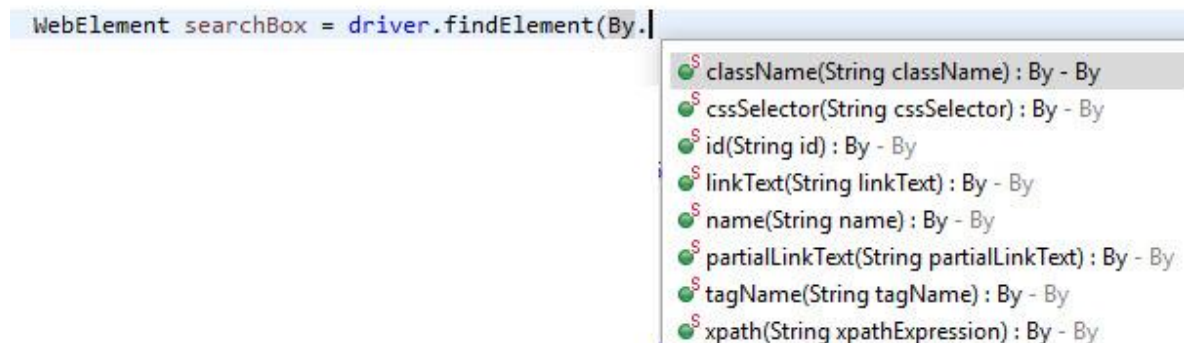


Ilustración 38. WebElement.

Cada WebElement tiene un nombre con un valor de atributo, todos estos localizadores deben ser únicos y descriptivos, no puede haber localizadores con el mismo nombre por qué esto impedirá en la automatización de pruebas que puedan ser localizados en el aplicativo, también se debe evitar que estos localizadores cambien ya que tendríamos que volver a cambiar toda la parte de los localizadores en las pruebas.

Podemos localizar los elementos por:

- Por identificación
- Nombre de la clase
- Nombre de etiqueta
- Nombre
- Enlace de texto
- CSS Selector
- XPath

Capítulo 4. Métodos para localizar elementos en la web de la clase By

| Variación | Descripción | Muestra |
|--------------------|---|---|
| By.className | Encuentra elementos basados en el valor del atributo «class» | findElement(By.className(«algúnnombreclase»)) |
| By.cssSelector | Encuentra elementos basados en el motor selector CSS subyacente del controlador | findElement(By.cssSelector(«input#email»)) |
| By.id | localiza los elementos por el valor de su atributo «ID» | findElement(By.id(«algunID»)) |
| By.linkText | Encuentra un elemento link con el texto exacto que muestra | findElement(By.linkText(«AlgunTexto»)) |
| By.name | Localiza los elementos por el valor del atributo «Name» | findElement(By.name(«AlgunName»)) |
| By.partialLinkText | Localiza elementos que contienen el texto de enlace dado | findElement(By.partialLinkText(«REG»)) |
| By.tagName | Localiza elementos por su nombre de etiqueta | findElement(By.tagName(«etiqueta»)) |
| By.xpath | Localiza elementos mediante XPath | findElement(By.xpath(«rutacpath»)) |

Nota sobre el uso de findElement (by, cssSelector ())

Ilustración 39. Web Element Clase By.

Capítulo 5. Metodología de desarrollo de software

La metodología scrum se utiliza en la empresa y es la que se está utilizando en el presente proyecto de pensiones, esta metodología ágil implica constante iteraciones, en la cual el código se modifica constantemente, en cada desarrollo o en cada iteración se implementan nuevas funcionalidades dentro del aplicativo, todas estas iteraciones involucran toda la parte de pruebas automatizadas donde ese necesario la ejecución de pruebas de regresión para poder verificar continuamente el funcionamiento de las diferentes funcionalidades. Esta metodología es incremental donde se dividen los requisitos y tareas en bloques de tiempo. Las etapas que se ven dentro de esta metodología planificación de la iteración (planning sprint), ejecución (sprint), reunión diaria (daily meeting) y demostración de resultados (sprint review). Cada iteración por estas etapas se denomina también sprint.

Capítulo 6. Resultados

6.1. Introducción

Antes de dar inicio con todo el ciclo de pruebas que se implementó para estas funcionalidades, se dará una pequeña introducción de ellas para posteriormente iniciar con el proceso de pruebas. Daremos la definición de unos conceptos que estaremos utilizando a lo largo de las siguientes funcionalidades:

- **Asegurado:** Persona natural que cuando inicia el proceso de afiliación a un fondo de pensiones se le crea el rol de afiliado o asegurado que nos indica que esta persona ya tiene una afiliación a este fondo.
- **Empleador:** Se puede definir como persona jurídica que tiene rol de empleador, en este caso aplica para empresas.

6.2. *Funcionalidad Registro de empleador*

El registro de solicitud de empleador se encuentra en el módulo de personas en la sección de empleadores en el aplicativo de Pensiones, este proceso nos permite registrar el proceso de creación de una persona Jurídica, en este caso una empresa; donde se realizan las acciones de adicionar un registro de solicitud de creación de un empleador, consultar la solicitud, editar la solicitud, aprobarla la solicitud de registro y consultar el histórico.

La sección de Registro de Solicitud de empleador contempla los siguientes casos de uso:

- CUPE1410 Gestionar solicitudes de registro de empleador
- CUPE1411 Crear solicitud de registro de empleador
- CUPE1412 Editar solicitud de registro de empleador
- CUPE1413 Crear empleador automáticamente

- CUPE1415 Crear persona automáticamente
- CUPE1416 Crear ubicación automáticamente

Lo primero que se hizo para el proceso del ciclo de pruebas fue la lectura de los requerimientos del cliente que están en estos casos de uso mencionados anteriormente.

Posteriormente se hicieron unas pruebas de interfaz para saber cómo estaba el aplicativo respecto a los requerimientos del cliente

| Identificación | Razón social | Estado | Número declaración | Fecha declaración | Acciones |
|----------------|--------------------|-----------|--------------------|-------------------|----------|
| 498759-2 | SOCIEDAD DE LA PAZ | Rechazada | 1652276009883 | 2022-05-11 | [Iconos] |
| 234 | PRUEBA MARI 2 | Rechazada | w23432 | 2022-05-07 | [Iconos] |
| 147859658 | EMPAQUES | Rechazada | 1652368904142 | 2022-05-06 | [Iconos] |

Ilustración 40. Funcionalidad Empleador. Elaboración Propia.

Se realizó la definición de los casos de prueba (DPC) o los diseños de plan de calidad donde se planeó todos los escenarios que se van a probar, para ello se generó cuatro casos de prueba que son los siguientes:

- PTL-GIN-003_Agregar Solicitud Desclasificación Militar.xlsx

| HEINSOHN BUSINESS TECHNOLOGY | | PTL-GINF-003 Versión formato: 2016/04/01 Desarrollado por Heinsohn Business Technology |
|--|---|--|
| DOCUMENTO PLAN DE PRUEBAS | | |
| Fecha_ | 25/02/2022 | |
| Título de la prueba / Escenario | Crear Solicitud de Desclasificación de asegurado militar | |
| Casos de uso | 1. CUPEH36 Crear solicitud de desclasificación de militar | |
| | 2. | |
| | 3. | |
| | 4. | |
| | 5. | |
| Clase / Metodo | Clase: AgregarSolicitudDesclaAseguradosTest01 | Metodo: agregarSolicitudDesclaAseguradosTest01 |
| Usuario que ejecuta la prueba | Danna Michelle Arévalo Hernández | |
| Escenario 1: Crear Solicitud de desclasificación | | |
| <i>Descripción: Permite crear una solicitud y que quede en estado aprobada</i> | | |
| Pasos a realizar | Resultados esperados | |
| Ejecuta en el navegador la URL: https://pensiones-ci.heinsohn.com.co:8081/#/ | Presenta la pantalla "Pensiones / Cesantías / Inversión" con los campos Usuario, Clave y Ingresar | |
| Ingrese los datos de entrada Usuario: selenium, Clave: selenium, haga clic en el botón "INGRESAR" | Presenta el menú que corresponde a la aplicación de Pensiones Obligatorias | |
| Seleccione el menú "Personas" | Despliega el submenú "Gestionar, Vision 360 Afiliado, RAE" | |
| Seleccionar el submenú Gestionar | Se presenta las card de "Personas" y "Solicitudes" | |
| Seleccionamos la card de Asegurados | Se presenta la pantalla con las solicitudes de asegurados | |
| Se le da click en el boton de "Agregar" | Se presenta la pantalla de Agregar Asegurados en el step de solicitud | |

Ilustración 41. DPC Agregar Solicitud Desclasificación Militar. Elaboración Propia.

| | |
|---|--|
| Ingresamos los siguientes datos: - Cite Ministerio: 91 - Fecha de Cite: 28/02/2022 - Fecha de recepción: 22/02/2022 - Cua: 567 - Gestor: 312-MARISOL BAUTISTA | Se presenta la step de Datos de Asegurado |
| Ingresamos los siguientes datos: - Tipo De Identificación: Carnet de extranjero (E) - Numero De Identificación: 4000500 - Fecha Nacimiento: 20/01/1969 - Primer Nombre: RODRIGO - SegundoNombre: ANDRES - PrimerApellido: GOMEZ - SegundoApellido: AMEZQUITA - Nacionalidad: BOLIVIA - PaisDeNacimiento: BOLIVIA - Ciudad: SUCRE / OROPEZA / CHUQUISACA - Genero: Masculino - EstadoCivil: SOLTERO(A) - Matricula: 690120GMR - CiudadUbicacion: COCHABAMBA / CERCADO / COCHABAMBA - Zona: Queru queru - Direccion: Av. Americas - Numero: 500 - Referencia: Pruebas - CorreoElectronico: pruebas@gestora.com.bo - TipoDeCorreoElectronico: Personal - Celular: 3202272833 Le damos click en el boton de siguiente | Se presenta la step de Validación documental |

| | |
|---|--|
| Ingresamos el documeto Nit solicitado | Se presenta el mensaje de éxito de guardado de documentos "Has adicionado exitosamente la solicitud de desclasificación de asegurado militar." |
| Le damos click en el boton de validar | |
| Le damos click en el boton de guardar | Se presenta el mensaje de éxito de guardad de solicitud "Has adicionado exitosamente la solicitud de desclasificación de asegurado militar." |
| Verificamos que haya quedado el registro guardado | |

Ilustración 42. DPC Adicionar Solicitud Desclasificación Militar. Elaboración Propia.

- PTL-GIN-003_Consultar Solicitud Desclasificación Militar.xlsx

| HEINSOHN BUSINESS TECHNOLOGY | | PTL-GINF-003 Versión formato: 2016/04/01 Desarrollado por Heinsohn Business Technology |
|--|---|--|
| DOCUMENTO PLAN DE PRUEBAS | | |
| Fecha | 25/02/2022 | |
| Título de la prueba / Escenario | Consultar Solicitud de Desclasificación de asegurado militar | |
| Casos de uso | 1. | CUPE1434 Gestionar solicitudes de desclasificación de militares |
| | 2. | |
| | 3. | |
| | 4. | |
| | 5. | |
| Clase / Metodo | Clase: ConsultarDesclaAseguradosTest01 | Metodo: consultarDesclaAseguradosTest01 |
| Usuario que ejecuta la prueba | Danna Michelle Arévato Hernández | |
| Escenario 1: Consultar Solicitud de desclasificación | | |
| <i>Descripción: Permite consultar una solicitud</i> | | |
| Pasos a realizar | | Resultados esperados |
| Ejecuta en el navegador la URL: https://pensiones-ci.heinsohn.com.co:8081/#/ | Presenta la pantalla "Pensiones / Cesantias / Inversion" con los campos Usuario, Clave y Ingresar | |
| Ingrese los datos de entrada Usuario: selenium, Clave: selenium, haga clic en el botón "INGRESAR" | Presenta el menú que corresponde a la aplicación de Pensiones Obligatorias | |
| Seleccione el menú "Personas" | Despliega el submenú "Gestionar, Vision 360 Afiliado, RAE" | |
| Seleccionar el submenu Gestionar | Se presenta las card de "Personas", "Solicitudes", "Asegurados" | |
| Seleccionamos la card de Asegurados | Se presenta la pantalla de consulta de solicitud | |
| Se le da click en el boton de consultar | Se presenta la pantalla de Adicionar Asegurados en el steper de solicitud | |

| | |
|--|--|
| Verificamos los siguientes datos: - Cite Ministerio: 91 - Fecha de Cite: 28/02/2022 - Fecha de recepción: 22/02/2022 - Cua: 567 - Gestor: MARISOL BAUTISTA - Estado: Observada - Observaciones: Pendiente documentación Matrícula diferente Le damos click en la card de datos de asegurado | Se presenta la card de Datos de Asegurado |
| Verificamos los siguientes datos: - Tipo De Identificación: Carnet de extranjero (E) - Numero De Identificación: 4000504 - Nombre Asegurado: RODRIGO GOMEZ - Fecha Nacimiento: 10/02/1993 - Nacionalidad: BOL - PaisDeNacimiento: BOLIVIA - Ciudad: CHAUNACA - Provincia: OROPEZA - Departamento: CHUQUISACA - Genero: MASCULINO - EstadoCivil: SOLTERO(A) - Matricula: 690120GMR - CiudadUbicacion: COCHABAMBA / CERCADO / COCHABAMBA - Zona: Queru queru - Direccion: Av. Americas - Numero: 500 - Referencia: Pruebas - CorreoElectronico: pruebas@gestora.com.bo - TipoDeCorreoElectronico: Personal - Celular: 3202272833 Le damos click en la card de validación documental | Se presenta la card de Validación documental |
| Verificamos el documento Nit Cargado | Se presenta la imagen o pdf |

Ilustración 43. DPC Consulta Desclasificación Asegurado Militar. Elaboración Propia.

- PTL-GIN-003_Filtros Solicitud Desclasificación Militar.xlsx

| | | |
|---------------------------------|--|--|
| HEINSOHN BUSINESS TECHNOLOGY | | PTL-GINF-003 Versión formato: 2016/04/01 Desarrollado por Heinsohn Business Technology |
| DOCUMENTO PLAN DE PRUEBAS | | |
| Fecha_ | 25/02/2022 | |
| Título de la prueba / Escenario | Filtros de Solicitud de Desclasificación de asegurado militar | |
| Casos de uso | 1. CUPE1434 Gestionar solicitudes de desclasificación de militares | |
| | 2. | |
| | 3. | |
| | 4. | |
| | 5. | |
| Clase / Metodo | Clase: FiltrarAseguradoTest01 | Metodo: filtrarAseguradoTest01 |
| Usuario que ejecuta la prueba | Danna Michelle Arévalo Hernández | |

| Escenario 1: Filtrar Solicitud de Desclasificación por Asegurado | |
|--|---|
| <i>Descripción: Permite filtrar una solicitud por asegurado</i> | |
| Pasos a realizar | Resultados esperados |
| Ejecuta en el navegador la URL: https://pensiones-ci.heinsohn.com.co:8081/#/ | Presenta la pantalla "Pensiones / Cesantías / Inversión" con los campos Usuario, Clave y Ingresar |
| Ingrese los datos de entrada Usuario: selenium, Clave: selenium, haga clic en el botón "INGRESAR" | Presenta el menú que corresponde a la aplicación de Pensiones Obligatorias |
| Seleccione el menú "Personas" | Despliega el submenú "Gestionar, Vision 360 Afiliado, RAE" |
| Seleccionar el submenú Gestionar | Se presenta las card de "Personas", "Solicitudes", "Asegurados" |
| Seleccionamos la card de Asegurados | Se presenta la pantalla de consulta de solicitud |
| Se le da click en el filtro de Asegurado | Se presentan los campos del filtro |
| Ingresamos los datos del filtro: - Tipo de identificación asegurado: Carnet de extranjero (E) - Numero de identificación asegurado: 4000400 Le damos click en boton de aplicar filtro | Se presenta la solicitud del asegurado que filtramos |
| Verificamos los campos que trajo la solicitud del asegurado: - TipoDeIdentificacion: E 4000400 - Nombre: MARIA CAMILA CASTRO BENAVIDEZ - Cua: 456 - Matricula: 965510CEM - Estado: Aprobada - Origen: INDIVIDUAL - FechaDeRecepcion: 2022-02-25 | Cerramos sesión |

Ilustración 44. DPC Filtros Solicitud Desclasificación de Asegurados Militares. Elaboración Propia.

- PTL-GIN-003_Generar Certificación.xlsx

| | | |
|---------------------------------|--|--|
| | HEINSOHN BUSINESS TECHNOLOGY | PTL-GINF-003 Versión formato: 2016/04/01 Desarrollado por Heinsohn Business Technology |
| | DOCUMENTO PLAN DE PRUEBAS | |
| Fecha | 25/02/2022 | |
| Título de la prueba / Escenario | Generar Certificación | |
| Casos de uso | 1. CUPEI434 Gestionar solicitudes de desclasificación de militares | |
| | 2. | |
| | 3. | |
| | 4. | |
| | 5. | |
| Clase / Metodo | Clase: GenerarCertificacionTest01 | Metodo: generarCertificacionTest01 |
| Usuario que ejecuta la prueba | Danna Michelle Arévalo Hernández | |

| Escenario 1: Generar certificación asegurado | |
|--|--|
| Descripción: Permite generar certificación a un asegurado militar | |
| Pasos a realizar | Resultados esperados |
| Ejecuta en el navegador la URL: https://pensiones-ci.heinsohn.com.co:8081/#/ | Presenta la pantalla "Pensiones / Cesantías / Inversión" con los campos Usuario, Clave y Ingresar |
| Ingrese los datos de entrada Usuario: selenium, Clave: selenium, haga clic en el botón "INGRESAR" | Presenta el menú que corresponde a la aplicación de Pensiones Obligatorias |
| Seleccione el menú "Personas" | Despliega el submenú "Gestionar, Vision 360 Afiliado, RAE" |
| Seleccionar el submenú Gestionar | Se presenta las card de "Personas", "Solicitudes", "Asegurados" |
| Seleccionamos la card de Asegurados | Se presenta la pantalla de consulta de solicitud |
| Se el boton de Generar Certificación | Se presentan todos los datos que se deben diligenciar |
| Ingresamos los siguientes datos - Cua: 567 - Matricula: 690120GMR - CiteMinisterio: 91 - TipoDeIdentificacion: Carnet de extranjero (E) - NumeroDeIdentificacion: 4000503 - NombresYApellidos: RODRIGO GOMEZ - FechaNacimiento: 20/01/1969 Le damos click en el boton de generar certificacion | Se presenta el mensaje modal de guardado con éxito y se descarga automáticamente la carta de certificación |
| Le damos click en el boton de Aceptar | Cerramos sesión |

Ilustración 45. Generar certificación militar. Elaboración Propia.

Una vez que tenemos el diseño de pruebas de calidad vamos a iniciar con la creación de los Scripts, donde tendremos el set de los datos necesarios para la ejecución de las pruebas automatizadas.

```

1  Insert into TPER_PERSONA (PERID, PERIDIPOIDENTIFICACION, PERCDNUMEROIDENTIFICACION, PERNMDIGITOVERIFICACION, PERDSPRIMERNOMBRE, PERDSSEI
2  Insert into TPER_EMPLEADOR (EMPID, EMPIDPERSONA, EMPSNINDICASIGRALPARTICIPA, EMPIDIPOCLASEAPORTANTE, EMPIDFUNCIONARIOGESTOR, EMPDSCANALATENCION, EM
3  Insert into TPER_ROL_PERSONA (TRSID, TRSIDTIPOROLPERSONA, TRSIDPERSONA, TRSFECREACION, TRSCDUSUARIOCREACION, TRSFEMODIFICACION, TRSCDUSUARIOMODIFICA
4  Insert into TPER_UBICACION (UBIID, UBIIDPERSONA, UBIIDCIUDAD, UBIDSNOMBREUBICACION, UBISNINDICACORRESEMPLEADO, UBIIDPAGADORA, UBIIDDEPURADORA, UBIDSTI
5  Insert into TPER_SUCURSAL_EMPLEADOR (SEMIID, SEMDSCANAL, SEMDSNUMERODECLARACION, SEMFEFECHADECLARACION, SEMDSTIPODECLARACION, SEMIDFUNCIONARIOGESTOR,
6  Insert into TPER_SOLICITUD_EMPLEADOR (SEMIID, SEMIDIPOIDENTIFICACION, SEMCDNUMEROIDENTIFICACION, SEMDSRAZONSOCIAL, SEMDSNUMERODECLARACION, SEMFEFECH
7  || TO_CLOB(q'[ARCIA", "apellidoCasada": "", "generoRepLegal": "MASCULINO", "indicadorCreacionEmpleador": "MANUAL", "tipoPersonaEmpleadorEnum": "JUR
8  || TO_CLOB(q'[RIZ", "claseUbicacionEnum": "ANBAS", "indicadorCorrespondenciaEmpleado": null, "lugarCorrespondenciaEnum": "CENTRALIZADA", "canalEnum
9  || TO_CLOB(q'[laseAportante": 1, "numeroTrabajadores": 10, "idSectorEconomico": 1, "idActividadEconomico": 1, "idPersonaEmpleador": null, "idPersona
10
11  COMMIT

```

Ilustración 46. Script Before. Elaboración Propia.

```

1  DELETE TPER_EMPLEADOR WHERE EMPIDPERSONA IN (SELECT PERID FROM TPER_PERSONA where PERCDNUMEROIDENTIFICACION = '-1234567890');
2  DELETE TPER_ROL_PERSONA WHERE TRSIDPERSONA IN (SELECT PERID FROM TPER_PERSONA where PERCDNUMEROIDENTIFICACION = '-1234567890');
3  DELETE TPER_UBICACION WHERE UBIIDPERSONA IN (SELECT PERID FROM TPER_PERSONA where PERCDNUMEROIDENTIFICACION = '-1234567890');
4  DELETE TPER_SUCURSAL_EMPLEADOR WHERE SEMIDPERSONA IN (SELECT PERID FROM TPER_PERSONA where PERCDNUMEROIDENTIFICACION = '-1234567890');
5  DELETE TPER_PERSONA WHERE PERCDNUMEROIDENTIFICACION = '-1234567890';
6  DELETE TPER_SOLICITUD_EMPLEADOR WHERE SEMIID='-99999';
7
8  COMMIT
9

```

Ilustración 47. Script After. Elaboración Propia.

Estos scripts se estarán ejecutando dentro de las pruebas automáticas en la parte de los test, ya que son necesarios dentro del proceso de la funcionalidad de empleadores.

Una vez se realizan los scripts que serán ejecutados dentro de los test, se empieza con la creación de los dto, pages y test para las pruebas automáticas, es importante crear una carpeta con el respectivo nombre de la funcionalidad que se está probando, dicha carpeta estará dividida en dto, page, test y utils (clase que tiene todas las utilidades necesarias) como vemos a continuación en la ilustración 48.

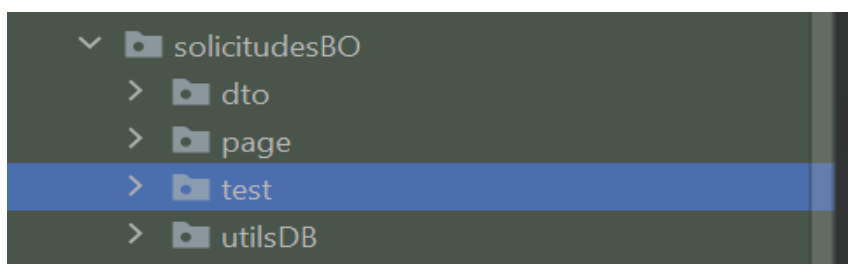


Ilustración 48. Carpetas Proyecto. Elaboración Propia.

Las primeras clases que se crearan son los dto, estas clases contienen todos los campos de creación de las pantallas necesaria para la creación de un empleador, cabe resaltar que los nombres de la carpeta van en minúscula y los nombres de clases que se creen la primera va en mayúscula, en estas clases dto estarán todos los campos de creación como vemos a continuación.

```
4  /**
5   * <b> Descripción:<b> Clase que determina los campos de creación de datos de empleador
6   * <b>Caso de uso:<b>
7   *
8   * @author Danna Michelle Arévalo Henández <darevalo@heinsohn.com.co>
9   * @version
10  *
11  */
12  public class AdicionarDatosEmpleadorDto {
13
14      private String tipoEmpleador;
15      private String oficina;
16      private String tipoDeIdentificacionEmple;
17      private String numeroDeIdentificacionEmple;
18      private String direccionAdministrativa;
19      private String codigoUnidadEjecutora;
20      private String razonSocial;
21      private String siglaNombreComercial;
22      private String fechaInicioActividad;
23      private String tipoDeEmpresa;
24      private String matriculaDeComercio;
25      private String sociedadComercial;
26      private String sectorEconomico;
27      private String actividadEconomica;
28      private String numeroDeTrabajadores;
29      private String sectorizacion;
```

Ilustración 49. Clase Dto Empleador. Elaboración Propia.

```

31  /**
32     * @return the tipoEmpleador
33     */
34  public String getTipoEmpleador() { return tipoEmpleador; }
37
38  /**
39     * @param tipoEmpleador the tipoEmpleador to set
40     */
41  public void setTipoEmpleador(String tipoEmpleador) { this.tipoEmpleador = tipoEmpleador; }
44
45  /**
46     * @return the oficina
47     */
48  public String getOficina() { return oficina; }
51
52  /**
53     * @param oficina the oficina to set
54     */
55  public void setOficina(String oficina) { this.oficina = oficina; }
58
59  /**
60     * @return the tipoDeIdentificacionEmple
61     */
62  public String getTipoDeIdentificacionEmple() { return tipoDeIdentificacionEmple; }
65

```

Ilustración 50. Clase Dto Empleador. Elaboración Propia.

Una vez tenemos las clases dto, se realiza las clases pages que contienen todas las acciones y los elementos que se tienen en pantalla, se tendrá un page para cada pantalla, dentro de estas clases se realizan los métodos para las acciones que vamos a realizar en pantalla, como podemos ver a continuación se tiene un método que ingresa todos los datos del empleador.


```

10  //**
11  * <b> Descripción:<b> Clase que contiene acciones y elementos de la pantalla de adicionar solicitud de
12  * empleador en la card de datos de empleador
13  * <b> Caso de uso:<b> CUPE1411
14  *
15  * @author Danna Michelle Arevalo Hernandez <darevalo@heinsohn.com.co>
16  * @version
17  *
18  */
19  public class AdicionarDatosEmpleadorPage {
20
21      private By tipoEmpleador = By.id("tipoEmpleador");
22      private By oficina = By.id("oficina");
23      private By tipoDeIdentificacionEmple = By.id("tipoIdentificacion");
24      private By numeroDeIdentificacionEmple = By.id("numeroIdentificacion");
25      private By direccionAdministrativa = By.id("txtDireccionAdministrativa");
26      private By codigoUnidadEjecutora = By.id("txtCodigoUnidadEjecutora");
27      private By razonSocial = By.id("razonSocial");
28      private By siglaNombreComercial = By.id("siglaNombreComercial");
29      private By fechaInicioActividad = By.id("fechaInicioActividadesInput");
30      private By tipoDeEmpresa = By.id("tipoEmpresa");
31      private By matriculaDeComercio = By.id("txtmatriculaComercio");
32      private By sociedadComercial = By.id("sociedadComercial");
33      private By sectorEconomico = By.id("sectorEconomico");
34      private By actividadEconomica = By.id("actividadEconomica");
35      private By numeroDeTrabajadores = By.id("txtnumeroTrabajadores");

```

Ilustración 51. Clase Page Empleador. Elaboración Propia.

```

41      private WebDriver driver;
42      private SeleniumWebDriverUtils utils;
43
44      public AdicionarDatosEmpleadorPage (WebDriver driver) {
45          this.driver = driver;
46          utils = new SeleniumWebDriverUtils(this.driver);
47      }
48

```

Ilustración 52. Clase Page Empleador. Elaboración Propia.

```

49  /**
50  *
51  * Metodo encargado de ingresar todos los datos del empleador
52  *
53  * @author Danna Michelle Arevalo Hernandez <darevalo@heinsonhn.com.co>
54  *
55  *
56  */
57  @ public void ingresarDatosEmpleador (AdicionarDatosEmpleadorDto adicionarDatosEmpleadorDto) {
58
59      utils.waitForOneSeconds();
60      utils.waitForOneSeconds();
61      utils.waitForOneSeconds();
62      utils.waitForOneSeconds();
63
64      // Seleccionamos el tipo de empleador
65      if (adicionarDatosEmpleadorDto.getTipoEmpleador() != null) {
66          utils.selectOptionList(tipoEmpleador, adicionarDatosEmpleadorDto.getTipoEmpleador());
67      }
68
69      utils.waitForOneSeconds();
70      utils.waitForOneSeconds();
71
72      // Seleccionamos el tipo de oficina
73      if (adicionarDatosEmpleadorDto.getOficina() != null) {
74          utils.selectOptionList(oficina, adicionarDatosEmpleadorDto.getOficina());

```

Ilustración 53. Método Clase Page Empleador. Elaboración Propia.

Las clases principales que se crearan a continuación son los test que se encargan de hacer todo el proceso dentro del aplicativo, en la primera parte de la clase test se hace las instancias sobre los pages que se creó, posteriormente dentro del método principal de la clase empieza todo el proceso de abrir el navegador, ingresar el usuario, contraseña, ingresar al módulo de personas y diligenciar todas las secciones del empleador para posteriormente guardar la solicitud del proceso y verla en pantalla, una vez termina el proceso, cierra la sesión y cierre el navegador donde se hace el proceso, se pondrá un ejemplo a continuación:

```

18 /**
19  * <b> Descripción:<b> Clase que permite adicionar una solicitud de empleador
20  * <b> Caso de uso:<b> CUPE1411
21  *
22  * @author Danna Michelle Arevalo Hernandez <darevalo@heinsohn.com.co>
23  * @version
24  *
25  */
26 public class AdicionarSolicitudEmpleadorTest01 {
27
28     private SeleniumTest selenium;
29     private LoginPage loginPage;
30     private MenuPage menuPage;
31     private SolicitudesEmpleadorPage solicitudesEmpleadorPage;
32     private AdicionarEmpleadorDeclaracionPage adicionarEmpleadorDeclaracionPage;
33     private AdicionarDatosEmpleadorPage adicionarDatosEmpleadorPage;
34     private AdicionarUbicacionPage adicionarUbicacionPage;
35     private AdicionarRepresentanteLegalPage adicionarRepresentanteLegalPage;
36     private AdicionarInfoNotarioPage adicionarInfoNotarioPage;
37     private EntrarCardPersonasUtils entrarCardPersonasUtils;
38
39     List<String> filaEncontrada;
40
41     /**
42     *
43     * Metodo encargado de implementar las instancias antes de la ejecucion de la prueba

```

Ilustración 54. Clase Test Adicionar Solicitud Empleador. Elaboración Propia.

```

70 /**
71  * Metodo encargado de crear una solicitud para un empleador
72  *
73  * @author Danna Michelle Arevalo Hernandez <darevalo@heinsohn.com.co>
74  *
75  */
76 @Test(groups = { "solicitudEmpleador" })
77 public void AdicionarSolicitudEmpleadorTest01() {
78
79     // Ingresar a Personas
80     loginPage.login(SeleniumProperties.getPropiedad(ConfigSeleniumPropiedades.USUARIO),
81         SeleniumProperties.getPropiedad(ConfigSeleniumPropiedades.CONTRASENA));
82     menuPage.clickOnMenu(menuPage.personas, menuPage.gestionar);
83
84     // En el modulo de gestionar entramos a la card de personas
85     entrarCardPersonasUtils.entrarCardSolicitudes();
86
87     // Le damos click en el boton de adicionar
88     solicitudesEmpleadorPage.clickAdicionar();
89
90     // Ingresamos los datos de la declaracion de empleador
91     AdicionarDeclaracionesDto adicionarDeclaracionesDto = new AdicionarDeclaracionesDto();
92     adicionarDeclaracionesDto.setCanal("Automático");
93     adicionarDeclaracionesDto.setFechaDeclaracion("24/01/2022");
94     adicionarDeclaracionesDto.setGestor("312-MARISOL BAUTISTA");
95

```

Ilustración 55. Método Clase Test Adicionar Solicitud Empleador. Elaboración Propia.

```

96     adicionarEmpleadoDeclaracionPage.declaracion(adicionarDeclaracionesDto);
97
98     // Ingresamos los datos del empleador
99     AdicionarDatosEmpleadoDto adicionarDatosEmpleadoDto = new AdicionarDatosEmpleadoDto();
100    adicionarDatosEmpleadoDto.setTipoEmpleado("Con dependientes");
101    adicionarDatosEmpleadoDto.setOficina("Casa matriz");
102    adicionarDatosEmpleadoDto.setTipoDeIdentificacionEmple("Número de Identificación Tributaria");
103    adicionarDatosEmpleadoDto.setNumeroDeIdentificacionEmple("1000456729");
104    adicionarDatosEmpleadoDto.setCodigoUnidadEjecutora("111");
105    adicionarDatosEmpleadoDto.setRazonSocial("PRUEBAS");
106    adicionarDatosEmpleadoDto.setSiglaNombreComercial("BANCO POPULAR");
107    adicionarDatosEmpleadoDto.setFechaInicioActividad("27/01/2022");
108    adicionarDatosEmpleadoDto.setTipoDeEmpresa("Mixta");
109    adicionarDatosEmpleadoDto.setMatriculaDeComercio("PBP");
110    adicionarDatosEmpleadoDto.setSociedadComercial("A- Empresa grande");
111    adicionarDatosEmpleadoDto.setSectorEconomico("AGRICULTURA, GANADERIA, CAZA Y ACTIVIDADES DE SERVICIOS CONEXAS");
112    adicionarDatosEmpleadoDto.setActividadEconomica("PRODUCCION ESPECIALIZADA DEL CAFE");
113    adicionarDatosEmpleadoDto.setNumeroDeTrabajadores("50");
114    adicionarDatosEmpleadoDto.setSectorizacion("Petroleros");
115
116    adicionarDatosEmpleadoPage.ingresarDatosEmpleado(adicionarDatosEmpleadoDto);

```

Ilustración 56. Método Clase Test Adicionar Solicitud Empleador. Elaboración Propia.

```

135    adicionarRepresentanteLegalDto.setTipoIdRepresentLegal("Carnet de extranjero");
136    adicionarRepresentanteLegalDto.setNumeroIdRepresentLegal("20203031");
137    adicionarRepresentanteLegalDto.setPrimerNombre("Edwin");
138    adicionarRepresentanteLegalDto.setSegundoNombre("Santiago");
139    adicionarRepresentanteLegalDto.setPrimerApellido("Prado");
140    adicionarRepresentanteLegalDto.setSegundoApellido("Benavidez");
141    adicionarRepresentanteLegalDto.setGenero("Masculino");
142
143    adicionarRepresentanteLegalPage.ingresarRepresentanteLegal(adicionarRepresentanteLegalDto);
144
145    //Ingresamos los datos del notario
146    AdicionarInfoNotarioDto adicionarInfoNotarioDto = new AdicionarInfoNotarioDto();
147    adicionarInfoNotarioDto.setNombreNotarioEmitePoder("Santiago");
148    adicionarInfoNotarioDto.setNumeroDeNotaria("29");
149    adicionarInfoNotarioDto.setNumeroDelPoderRepresentLegal("95");
150    adicionarInfoNotarioDto.setFechaEmisionPoderRepresentLegal("27/01/2022");
151
152    adicionarInfoNotarioPage.ingresarInfoNotario(adicionarInfoNotarioDto);
153
154    adicionarInfoNotarioPage.cargarArchivo();
155
156    //Cerramos sesion
157    loginPage.logout();
158 }

```

Ilustración 57. Método Clase Test Adicionar Solicitud Empleador. Elaboración Propia.

```

160 /**
161  *
162  * Metodo encargado de cerrar la prueba
163  *
164  * @author Danna Michelle Arevalo Hernandez <darevalo@heinsohn.com.co>
165  *
166  */
167 @AfterMethod(groups = { "solicitudEmpleador" })
168 public void cerrar(){
169     selenium.getDriver().close();
170 }
171
172 }
173

```

Ilustración 58. AfterMethod Clase Test Adicionar Solicitud Empleador. Elaboración Propia.

Una vez que tenemos todos los test con los escenarios planeados, se lanzan todas las pruebas para obtener el reporte de dichas pruebas y saber si fueron satisfactorias, para ello abrimos símbolo del sistema y ejecutamos el siguiente comando:

```

C:\Windows\System32\cmd.exe
Microsoft Windows [Versión 10.0.19043.1526]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\obligatorias\selenium-padre\personas-selenium>mvn test -fae -Pselenium -Dgroups=solicitudEmpleador -Dmaven.test.skip=false

```

Ilustración 59. Comando Ejecución de Pruebas en Grupo. Elaboración Propia.

Desde la carpeta donde tenemos las pruebas ejecutamos el comando que se en la ilustración con el nombre del grupo de pruebas que vamos a correr que en este caso es solicitudEmpleador, una vez que lo ejecutemos se correrán las pruebas automáticas que estén dentro de ese grupo, cuando termina de correr todas las pruebas automáticas, se genera un reporte de pruebas que podemos consultarlo en la carpeta donde tenemos las pruebas en este caso personas-selenium > target > surefire-reports > personasInfoBasica > index.html.

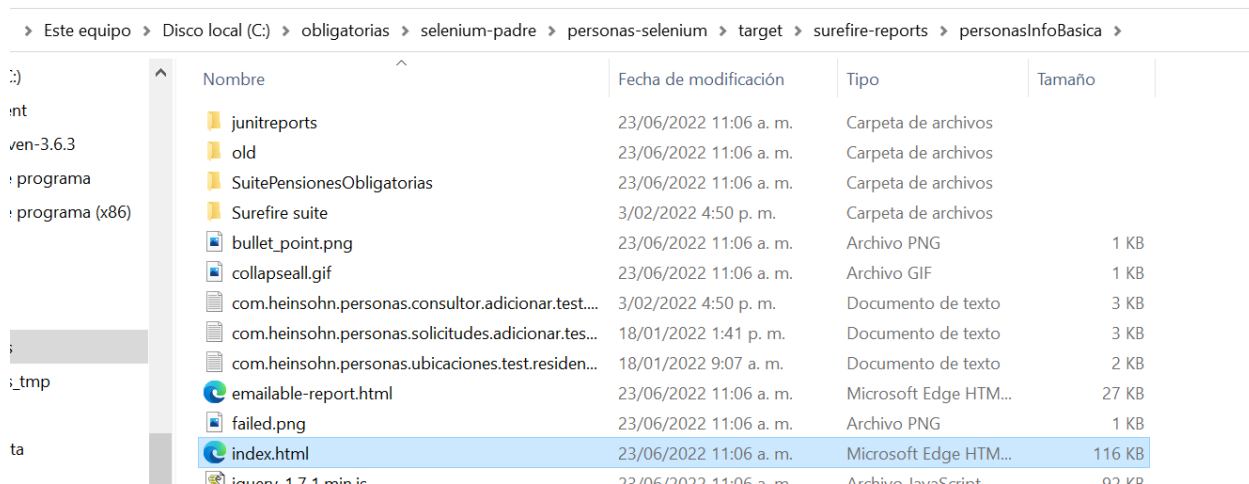


Ilustración 60. Carpeta Reporte Pruebas Selenium. Elaboración Propia.

Una vez ingresamos a `index.html` en el navegador se abrirá una ventana con todos los reportes de las pruebas que se corrieron dentro del grupo, donde podemos ver si fueron satisfactorias.

| Test method | Exception | Time (seconds) | Instance |
|---|-----------|----------------|--|
| AdicionarSolicitudEmpleador Test01 Test class: com.heinsohn.personas.solicitudes.adicionar.test.AdicionarSolicitudEmpleadorTest01 | | 205 | com.heinsohn.personas.solicitudes.adicionar.test.AdicionarSolicitudEmpleadorTest01@... |
| AdicionarSolicitudEmpleador Test01 Test class: com.heinsohn.personas.solicitudes.adicionar.test.AdicionarSolicitudEmpleadorTest03 | | 132 | com.heinsohn.personas.solicitudes.adicionar.test.AdicionarSolicitudEmpleadorTest03@... |
| AdicionarSolicitudEmpleador Test02 Test class: com.heinsohn.personas.solicitudes.adicionar.test.AdicionarSolicitudEmpleadorTest02 | | 94 | com.heinsohn.personas.solicitudes.adicionar.test.AdicionarSolicitudEmpleadorTest02@... |
| AdicionarSolicitudEmpleador Test04 Test class: com.heinsohn.personas.solicitudes.adicionar.test.AdicionarSolicitudEmpleadorTest04 | | 126 | com.heinsohn.personas.solicitudes.adicionar.test.AdicionarSolicitudEmpleadorTest04@... |

Ilustración 61. Reporte de ejecución de Pruebas Desclasificación de Asegurados Militares. Elaboración Propia.

Una vez que se dan por terminado el proceso de pruebas automatizadas, se hace la entrega al cliente y se programa una reunión para mostrar todo el proceso.

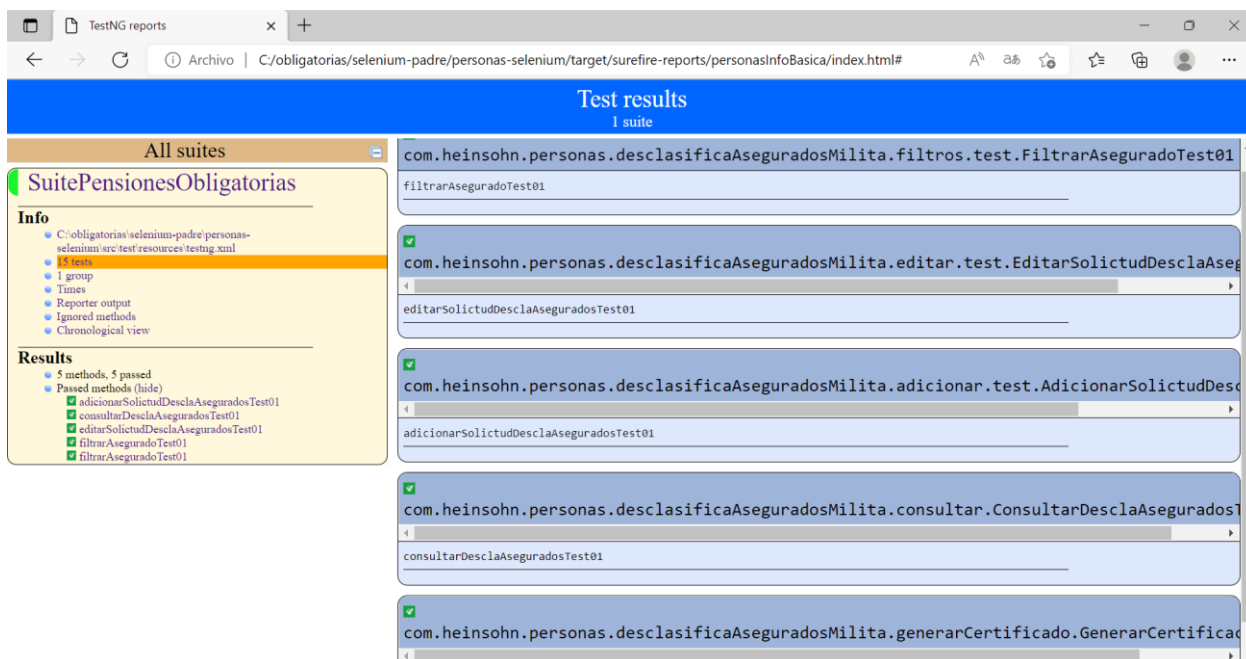


Ilustración 62. Reporte de ejecución de Pruebas Desclasificación de Asegurados Militares. Elaboración Propia.

6.3. Funcionalidad Registro de Consultor

El registro de solicitud de consultores se encuentra en el módulo de personas en la sección empleadores, esta funcionalidad nos permite asignarle a una persona natural el rol de consultor, el sistema nos permite adicionar una solicitud de registro de consultor, consultarla, editarla, aprobarla y consultar el histórico de dicha solicitud.

Para el inicio de las pruebas de la funcionalidad de registro de consultor, en primera instancia se hace lectura de los casos de uso para mirar los requerimientos del cliente, los casos de uso de esta sección corresponden a:

- PT-INGE-014-CUPE1418 BOL Crear solicitud de registro de consultor
- PT-INGE-014-CUPE1417 BOL Gestionar solicitudes de registro de consultor
- PT-INGE-014-CUPE1419 BOL Editar solicitud de registro de consultor
- PT-INGE-014-CUPE1420 BOL Crear consultor y contratante automáticamente

Una vez se realiza la lectura de los casos de uso, se empieza con el proceso de pruebas de interfaz, donde se miraba todas las funcionalidades de registro de consultor para ver su funcionamiento antes del inicio de las pruebas automáticas, también se verifica que todos los objetos dentro de la pantalla cuenten con el localizador id para poder realizar las pruebas.

The screenshot displays the 'Empleador / FRC' interface. At the top, there are three navigation buttons: 'Empleador', 'FRC' (highlighted in red), and 'FDNIR'. Below these is the title 'Formulario Registro Consultor' and four filter dropdowns: 'Contratante', 'Estado', 'Fecha declaración', and 'No. declaración'. A '+ ADICIONAR' button is located on the right. The main content is a table with the following data:

| Identificación contratante | Razón social | Estado | Número declaración | Fecha declaración | Acciones |
|----------------------------|----------------------|----------|--------------------|-------------------|---|
| NIT 1023658974 | SanitasP | Grabada | 1655393679828 | 16/06/2022 | [Icons for search, edit, delete, share] |
| NIT 102587969 | LSCAFAM | Grabada | 1654647062173 | 02/06/2022 | [Icons for search, edit, delete, share] |
| NIT 148478478 | PAPERMATE | Grabada | 1653521320354 | 25/05/2022 | [Icons for search, edit, delete, share] |
| NIT 4370332016 | RENE MAYTA QUIUCHACA | Aprobada | 14528 | 25/05/2022 | [Icons for search, edit, delete, share] |

Ilustración 63. Funcionalidad Formulario Registro de Solicitud de Consultores. Elaboración Propia.

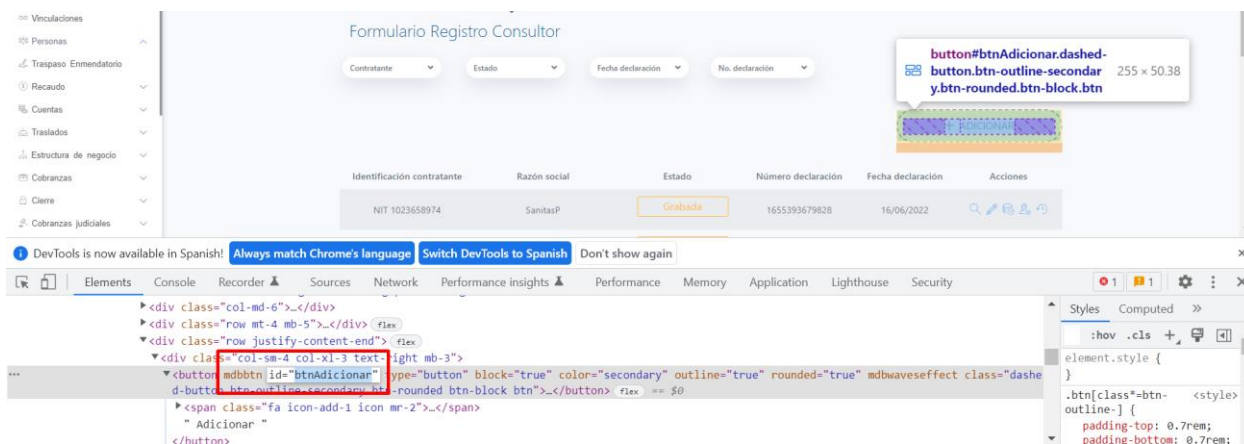


Ilustración 64. Localizador id. Elaboración Propia.

Terminadas las pruebas de interfaz y rectificado que todos los localizadores necesarios para realizar las pruebas automáticas están, se inicia con el proceso de diseño de pruebas de calidad (DPC).

- Adicionar Solicitud de Registro de Consultor

| | | | |
|----|--|--|---|
| 1 | | HEINSOHN BUSINESS TECHNOLOGY | PTL-GINF-003 Versión formato: 2016/04/01 Desarrollado por Heinsohn Business Technology |
| 2 | | DOCUMENTO PLAN DE PRUEBAS | |
| 3 | | | |
| 4 | Fecha_ | | 4/02/2022 |
| 5 | Título de la prueba / Escenario | Crear Solicitud Consultor | |
| 6 | | 1. | CUPE1418 Crear solicitud de registro del consultor |
| 7 | | 2. | |
| 8 | Casos de uso | 3. | |
| 9 | | 4. | |
| 10 | | 5. | |
| 11 | Clase / Metodo | Clase: AdicionarConsultorTest02 | Metodo: AdicionarConsultorTest02 |
| 12 | Usuario que ejecuta la prueba | Danna Michelle Arévalo Hernández | |
| 13 | | | |
| 14 | Escenario 1: Crear Solicitud de consultor | | |
| 15 | <i>Descripción: Permite crear una solicitud de consultor</i> | | |
| 16 | | Pasos a realizar | Resultados esperados |
| 17 | | Ejecuta en el navegador la URL: https://pensiones-ci.heinsohn.com.co:8081/#/ | Presenta la pantalla "Pensiones / Cesantías / Inversión" con los campos Usuario, Clave y Ingresar |
| 18 | | Ingrese los datos de entrada Usuario: selenium, Clave: selenium, haga clic en el botón "INGRESAR" | Presenta el menú que corresponde a la aplicación de Pensiones Obligatorias |
| 19 | | Seleccione el menú "Personas" | Despliega el submenú "Gestionar, Vision 360 Afiliado, RAE" |
| 20 | | Seleccione el submenú Gestionar | Se presenta las card de "Personas" y "Solicitudes" |
| 21 | | Seleccionamos la card de Solicitudes | Se presenta la pantalla de Solicitudes con las card de Empleador y FRC |

Ilustración 65. DPC Adicionar Solicitud de Registro de Consultor. Elaboración Propia.

| | | |
|----|---|---|
| 22 | Seleccionamos la card de FRC | Se presenta la pantalla de FRC con todas las acciones de solicitud de consultor |
| 23 | Se le da click en el boton de "Adicionar" | Se presenta la pantalla de Adicionar consultor |
| 24 | Ingresamos los datos de solicitud de declaración del primer steper: - Canal: Automático - Fecha de declaración: 04/02/2022 - Documentos presentados: Impreso - ¿Se constituye agente de retención? : No - Gestor: 312-MARISOL BAUTISTA | Se presenta la steper de Datos de datos de contratante |
| 24 | Ingresamos los datos del contratante: - Tipo de identificación: GOB - Número de identificación: 6000200 - Dirección administrativa: 009092 - Código unidad ejecutora: 1 Desplegamos los datos de ubicación y ingresamos: - Ciudad: COCHABAMBA / SECCION CAPITAL / CERCADO - Zona: Queru queru - Dirección: Av. Americas - Número: 421 - Referencia: Cochabamba - Correo: nutresa@bolivia.com.bo - Tipo de correo electrónico: Laboral - Celular: 3206744244 - Telefono: 3759000 - Extensión: 2 | Se presenta de representante legal |
| 25 | | |

Ilustración 66. DPC Adicionar Solicitud de Registro de Consultor. Elaboración Propia.

| | | |
|----|--|---|
| 26 | Ingresamos datos del representante legal: - Tipo de identificación: Carnet de extranjero - Numero de identificación: 90122000 - Primer nombre: JAIME - Segundo nombre: RICARDO - Primer apellido: FERNANDEZ - Segundo apellido: VAQUERO - Genero: Masculino - Ciudad: SUCRE / SECCION CAPITAL / OROPEZA - Zona: Pruebas - Dirección: Calle 22 A - Numero: 29-30 - Referencia: OROPEZA - Correo electronico: ricardo@bolivia.com.bo - Tipo de correo: Personal - Celular: 3047888888 | Se presenta la steper de consultores |
| 27 | Le damos click en el boton de "ADICIONAR" consultor | Se presenta la pantalla con los campos de Adicionar Consultor |

Ilustración 67. DPC Adicionar Solicitud de Registro de Consultor. Elaboración Propia.

| | |
|---|--|
| <p>Adicionamos los datos de consultor: - Tipo de identificación: Carnet de extranjero - Numero de identificación: 90122001</p> <p>Una vez ingresados el tipo de identificación y Número de identificación nos cargan los campos de:</p> <ul style="list-style-type: none"> - CUA - Primer nombre - Segundo nombre - Primer apellido - Segundo Apellido <p>Ingresamos los otros datos faltantes</p> <ul style="list-style-type: none"> - Departamento: SUCRE - Tipo consultor: Consultor - Monto total del contrato: 900.000 - Fecha inicio del contrato: 12/01/2022 - Fecha fin del contrato: 04/02/2022 - Fecha inicio adeuda al contrato: 12/01/2022 - Fecha fin adeuda al contrato: 04/02/2022 | <p>Se presenta el mensaje de exitoso guardado de consultor "Has adicionado exitosamente el asegurado."</p> |
| 28 | <p>Se presenta la pantalla de adicionar consultor con el registro de consultor agregado</p> |
| 29 | <p>Le damos click en el boton de "ACEPTAR"</p> |
| 30 | <p>Le damos click en el boton de "SIGUIENTE"</p> |
| 31 | <p>Subimos el DOCUMENTO DE IDENTIDAD NIT obligatorio</p> |
| 32 | <p>Le damos click en boton de "VALIDAR"</p> |
| 33 | <p>Le damos click en boton de "ACEPTAR"</p> |
| 33 | <p>Le damos click en boton de "GUARDAR"</p> |
| 34 | <p>Le damos click en boton de "ACEPTAR"</p> |
| 35 | <p>Cerramos la sesión y la pagina</p> |

Ilustración 68. DPC Adicionar Solicitud de Registro de Consultor. Elaboración Propia.

- Filtros Solicitudes Registro Consultor

| | | | |
|----|--|---|--|
| 1 | HEINSOHN BUSINESS TECHNOLOGY | | PTL-GINF-003 |
| 2 | DOCUMENTO PLAN DE PRUEBAS | | Versión formato: 2016/04/01 |
| 3 | | | Desarrollado por Heinsohn Business Technology |
| 4 | Fecha | 31/01/2022 | |
| 5 | Título de la prueba / Escenario | Validar gestionar solicitud consultor | |
| 6 | | 1. CUPE1417 Gestionar solicitudes de registro consultor | |
| 7 | | 2. CUPE1418 Crear solicitud de registro de consultor | |
| 8 | Casos de uso | 3. CUPE1419 Editar solicitud de registro consultor | |
| 9 | | 4. | |
| 10 | | 5. | |
| 11 | Usuario que ejecuta la prueba | Danna Michelle Arévalo Hernández | |
| 12 | Clase / Metodo | Clase:FiltrosSolicitudConsultoresEsc01 | Metodo:filtrosSolicitudConsultorEsc01 |
| 13 | | | |
| 14 | Escenario 1: Verificar el funcionamiento de los diferentes filtros | | |
| 15 | Descripción: Verificar que en la pantalla de solicitudes consultor funcionen de manera correcta los filtros de contratante, estado, fecha de declaracion y numero de declaracion | | |
| 16 | Pasos a realizar | | Resultados esperados |
| 17 | Ingresar al aplicativo de la gestora | | Se debe mostrar la pantalla principal de la gestora |
| 18 | Seleccionar el menu Personas | | Se despliega el submenu "Gestionar" |
| 19 | Seleccionar el submenu Gestionar | | Se muestra la pantalla de gestionar |
| 20 | Seleccionar Consultor | | Se muestra la pantalla del consultor con sus filtros y la grilla con los datos de las solicitudes |
| 21 | Seleccionar el filtro contratante | | Se despliega el filtro con tipo y numero de identificacion |
| | Ingresar los datos | | Se visualiza los campos ingresados en pantalla |
| | Tipo de identificacion:Nit | | |
| | numero de identificacion:900213886 | | |
| 22 | Seleccionar el boton aplicar | | Se muestra en la grilla la solicitud que se encontro con el tipo y numero de identificacion filtrado |
| 23 | Seleccionar el filtro estado | | Se despliega las opciones de estado |
| 24 | Seleccionar el estado "rechazada" | | Se visualiza que la opcion de rechazada quedo seleccionada |
| 25 | Seleccionar el boton aplicar | | Se muestra en la grilla la solicitud que se encontro con el estado filtrado |
| 26 | Seleccionar el filtro fecha de declaracion | | Se despliega el filtro para ingresar la fecha de inicio y la fecha fin de la declaracion |
| | Ingresar las fechas de la declaracion | | Se visualiza las fechas de declaracion ingresadas |
| | fecha de inicio: 01/02/2022 | | |
| | fecha de fin:01/02/2022 | | |
| 27 | Seleccionar el boton aplicar | | Se muestra en la grilla la solicitud que se encontro |
| 28 | Seleccionar el filtro de numero de declaracion | | Se despliega el filtro para ingresar el numero de declaracion |
| 29 | Ingresar el numero de declaracion : 12346 | | Se visualiza el numero de declaracion que se ingreso |
| 30 | Seleccionar el boton aplicar | | Se muestra en la grilla la solicitud que se encontro con el numero de declaracion |
| 31 | Seleccionar el boton aplicar | | Se muestra en la grilla la solicitud que se encontro con el numero de declaracion |
| 32 | Se valida la informacion de la grilla | | La solicitud que se muestra es la que se espera al hacer los filtros |
| 33 | Se selecciona el filtro contratante y se borra el filtro | | Se muestra el filtro de contratante sin filtro |
| 34 | Se selecciona el filtro estado y se borra el filtro | | Se muestra el filtro de estado sin filtro |
| 35 | Se selecciona el filtro fecha de declaracion y se borra el filtro | | Se muestra el filtro de fecha de declaracion sin filtro |
| 36 | Se selecciona el filtro numero de declaracion y se borra el filtro | | Se muestra el filtro de numero de declaracion sin filtro |
| 37 | Se selecciona el filtro fecha de declaracion | | Se despliega el filtro de fecha para ingresar las fechas. |
| 38 | Ingresar las fechas de la declaracion | | Se visualiza las fechas de declaracion ingresadas |
| | fecha de inicio: 01/02/2022 | | |
| | fecha de fin:01/03/2022 | | |
| 39 | Seleccionar el boton aplicar | | Se muestra en la grilla las solicitudes que se encontro |
| 40 | | | |
| 41 | | | |

Ilustración 69. DPC Filtros Solicitud de Registro de Consultor. Elaboración Propia.

- Histórico Solicitud de Registro de Consultor

| 1 Escenario 2: Verificar que al consultar el historial al hacer cambios en una solicitud muestre los registros de cambio | |
|---|---|
| 2 Descripción: Verificar que al realizar un cambio en una solicitud en estado GRABADO se visualice el registro en el historial, además se pueda filtrar por fecha y así poder comparar los dos registros y consultar el detalle del historico | |
| 3 Clase: HistoricoSolicitudConsultorEsc01 | |
| Metodo: revisarHistoricoConsultorEsc01 | |
| 4 Pasos a realizar | Resultados esperados |
| 5 Ingresar al aplicativo de la gestora | Se debe mostrar la pantalla principal de la gestora |
| 6 Seleccionar el menu Personas | Se despliega el submenu "Gestionar" |
| 7 Seleccionar el submenu Gestionar | Se muestra la pantalla de gestionar |
| 8 Seleccionar Consultor | Se muestra la pantalla del consultor con sus filtros y la grilla con los datos de las solicitudes |
| 9 En la grilla seleccionar el boton historico | Se muestra la pantalla del historico de la solicitud |
| 10 Seleccionar el filtro rango de fecha | Se despliega el filtro para ingresar las fechas |
| Ingresar las fechas de la declaracion fecha de inicio: 18/01/2022 | Se visualiza las fechas de declaracion ingresadas |
| 11 fecha de fin: 18/01/2022 | |
| 12 Seleccionar el boton aplicar | Se muestra en la grilla las solicitudes que se encontro |
| 13 | Se valida que el numero de registros esperados sea 2 |
| 14 | Se valida la informacion que el filtro trajo de las solicitudes |
| 15 Chequear los dos registros de solicitud | Se visualiza que los registros fueron seleccionados |
| 16 Seleccionar el boton comparar | Se muestra la pantalla del historico con la comparacion de los registros |
| 17 | Se valida la informacion que trae el comparar solicitudes |
| 18 Seleccionar el link volver | Se muestra la pantalla del historico |
| 19 En la grilla seleccionar el boton consultar | Se muestra la pantalla del detalle del historico |
| 20 | Se valida la informacion del detalle del historico |

Ilustración 70. DPC Escenario Uno Histórico Solicitud de Registro de Consultor. Elaboración Propia.

| 1 Escenario 3: Verificar que al consultar el historial al hacer cambios en una solicitud salgan datos cuando esta en estado rechazada | |
|---|---|
| 2 Descripción: Verificar que al realizar un cambio en una solicitud en estado RECHAZADA se visualice el registro en el historial, además se pueda filtrar por fecha y así poder comparar los dos registros y consultar el detalle del historico | |
| 3 Clase: HistoriaSolicitudesEsc02 | |
| Metodo: revisarHistoricoEsc02 | |
| 4 Pasos a realizar | Resultados esperados |
| 5 Ingresar al aplicativo de la gestora | Se debe mostrar la pantalla principal de la gestora |
| 6 Seleccionar el menu Personas | Se despliega el submenu "Gestionar" |
| 7 Seleccionar el submenu Gestionar | Se muestra la pantalla de gestionar |
| 8 Seleccionar Consultor | Se muestra la pantalla del consultor con sus filtros y la grilla con los datos de las solicitudes |
| 9 En la grilla seleccionar el boton historico | Se muestra la pantalla del historico de la solicitud |
| 10 Seleccionar el filtro rango de fecha | Se despliega el filtro para ingresar las fechas |
| Ingresar las fechas de la declaracion fecha de inicio: 20/01/2022 | Se visualiza las fechas de declaracion ingresadas |
| 11 fecha de fin: 20/01/2022 | |
| 12 Seleccionar el boton aplicar | Se muestra en la grilla las solicitudes que se encontro |
| 13 | Se valida que el numero de registros esperados sea 2 |
| 14 En la grilla seleccionar el boton consultar | Se muestra la pantalla del detalle del historico |
| 15 | Se valida la informacion del detalle del historico |

Ilustración 71. DPC Escenario Dos Histórico Solicitud de Registro de Consultor. Elaboración Propia.

- Rechazar Solicitud de Registro de Consultor

| 1 Escenario 4: Validar que se pueda rechazar una solicitud | |
|---|---|
| 2 Descripción: Validar que se pueda rechazar una solicitud que se encuentre en estado grabada | |
| 3 Clase: AprobarSolicitudConsultorEsc01 | |
| Metodo: aprobarSolicitudEsc01 | |
| 4 Pasos a realizar | Resultados esperados |
| 5 Ingresar al aplicativo de la gestora | Se debe mostrar la pantalla principal de la gestora |
| 6 Seleccionar el menu Personas | Se despliega el submenu "Gestionar" |
| 7 Seleccionar el submenu Gestionar | Se muestra la pantalla de gestionar |
| 8 Seleccionar Consultor | Se muestra la pantalla del consultor con sus filtros y la grilla con los datos de las solicitudes |
| 9 En la grilla seleccionar el boton de aprobacion | Se muestra el modal de aprobacion |
| 10 Selecciona la opcion de Rechazar | Se visualiza que la opcion de rechazar queda seleccionada |
| 11 Selecciona el boton de confirmar | Se muestra en pantalla el modal de actualizacion con el mensaje de "Has rechazado exitosamente la solicitud de registro del consultor." |
| 12 | Se valida la informacion de la grilla sobre esa solicitud |
| 13 | Se valida que el estado de la solicitud haya cambiado |
| 14 | |
| 15 | |

Ilustración 72. DPC Rechazar Solicitud de Registro de Consultor. Elaboración Propia.

- Validar Solicitud de Registro de Consultor

| | | |
|----|--|--|
| 1 | Escenario 5: Validar que se pueda aprobar una solicitud | |
| 2 | Descripción: Validar que se pueda aprobar una solicitud que se encuentre en estado grabada | |
| 3 | Clase: AprobarSolicitudConsultorEsc02 | Metodo: aprobarSolicitudEsc02 |
| 4 | Pasos a realizar | Resultados esperados |
| 5 | Ingresar al aplicativo de la gestora | Se debe mostrar la pantalla principal de la gestora |
| 6 | Seleccionar el menu Personas | Se despliega el submenu "Gestionar" |
| 7 | Seleccionar el submenu Gestionar | Se muestra la pantalla de gestionar |
| 8 | Seleccionar Consultor | Se muestra la pantalla del consultor con sus filtros y la grilla con los datos de las solicitudes |
| 9 | En la grilla seleccionar el boton de aprobacion | Se muestra el modal de aprobacion |
| 10 | Selecciona la opcion de Aprobar | Se visualiza que la opcion de aprobar queda seleccionada |
| 11 | Selecciona el boton de confirmar | Se muestra en pantalla el modal de actualizacion con el mensaje de "Has aprobado exitosamente la solicitud de registro del consultor." |
| 12 | | Se valida la informacion de la grilla sobre esa solicitud |
| 13 | | Se valida que el estado de la solicitud haya cambiado |
| 14 | | Se valida que se cree la persona en la entidad tper_persona con los datos de la solicitud |
| 15 | | Se valida que se cree la sucursal en la entidad tper_sucursal_empleador |
| 16 | | Se valida que la ubicacion se cree en la entidad tper_ubicacion |

Ilustración 73. DPC Validar Solicitud de Registro de Consultor. Elaboración Propia

- Editar Solicitud de Registro de Consultor

| | | |
|----|--|---|
| 1 | Escenario 6: Validar la edicion de una solicitud | |
| 2 | Descripción: Validar que una solicitud se pueda editar y que los campos que correspondan esten habilitados | |
| 3 | Clase: EditarSolicitudEsc01 | Metodo: modificarSolicitudEsc01 |
| 4 | Pasos a realizar | Resultados esperados |
| 5 | Ingresar al aplicativo de la gestora | Se debe mostrar la pantalla principal de la gestora |
| 6 | Seleccionar el menu Personas | Se despliega el submenu "Gestionar" |
| 7 | Seleccionar el submenu Gestionar | Se muestra la pantalla de gestionar |
| 8 | Seleccionar Consultor | Se muestra la pantalla del consultor con sus filtros y la grilla con los datos de las solicitudes |
| 9 | En la grilla seleccionar el boton de editar | Se muestra la pantalla de editar la solicitud con el primer stepper "declaracion" |
| 10 | Se da click sobre la opcion canal edita el canal y se selecciona la opcion de canal preimpreso | Se visualiza que la opcion del canal fue cambiada |
| 11 | Se da click sobre la opcion numero identificacion y se edita el numero de declaracion | Se visualiza que la opcion del numero de declaracion fue cambiada |
| 12 | | Se valida que el campo numero de declaracion solo pueda ser editado cuando el canal es de tipo preimpreso |
| 13 | Seleccionar el boton siguiente | Se presenta la pantalla de editar datos empleador |
| 14 | Se da click sobre la opcion razon social se edita la razon | Se visualiza que la opcion del numero de declaracion fue actualizada |
| 15 | Se da click sobre la opcion nombre comercial se edita el nombre | Se valida el campo de razon social solo se pueda editar cuando el empleador no existe |
| 16 | Se da click sobre la opcion matricula se edita la matricula | Se visualiza que la opcion del canal fue actualizada |
| 17 | Se da click sobre la opcion sociedad comercial se edita la sociedad | Se visualiza que la opcion de la matricula fue actualizada |
| 18 | Se da click sobre el campo numero de identificacion se edita el numero | Se visualiza que la opcion de la sociedad fue actualizada |
| 19 | | Se visualiza que la opcion del numero de identificacion fue actualizada |
| 20 | | Se verifica que el campo oficina no sea editable |
| 21 | | Se verifica que el campo tipo empresa cuando el tipo de identificacion sea igual nit sea editable |
| 22 | | Se verifica que el campo numero de trabajadores cuando sea igual a sea editable |
| 23 | Seleccionar el boton siguiente | Se presenta la pantalla de editar ubicacion |
| 24 | Se da click sobre la opcion lugar de correspondencia se edita el lugar | Se visualiza que la opcion del lugar de correspondencia fue actualizada |
| 25 | Se da click sobre el campo zona se edita la zona | Se visualiza que la opcion del lugar de correspondencia fue actualizada |
| 26 | Se da click sobre el campo direccion se edita la direccion | Se visualiza que la zona fue actualizada |
| 27 | Se da click sobre el campo coordenada de latitud se edita la coordenada | Se visualiza que la direccion fue actualizada |
| 28 | Se da click sobre el campo correo electronico se edita el correo | Se visualiza que la coordenada fue actualizada |
| 29 | Se da click sobre el campo celular se edita el celular | Se visualiza que el campo del correo fue actualizado |
| 30 | Se da click sobre el campo telefono electronico se edita el telefono | Se visualiza que el campo del celular fue actualizado |
| 31 | Se da click sobre el campo extencion se edita la extencion | Se visualiza que el campo del telefono fue actualizado |
| 32 | Seleccionar el boton siguiente | Se visualiza que la campo de la extencion fue actualizada |
| 33 | Se da click sobre la opcion tipo identificacion se edita el tipo | Se presenta la pantalla de editar representante legal |
| 34 | Se da click sobre el campo numero de identificacion se edita el numero | Se visualiza que la opcion del tipo de identificacion fue actualizada |
| 35 | Se da click sobre el campo primer nombre se edita el nombre | Se visualiza que la campo de la numero de identificacion fue actualizada |
| 36 | Se da click sobre el campo primer apellido se edita el apellido | Se visualiza que la campo del primer nombre fue actualizado |
| 37 | Se da click sobre la opcion genero se edita el genero | Se visualiza que la campo del primer apellido fue actualizado |
| 38 | Seleccionar el boton siguiente | Se visualiza que la opcion del genero fue actualizado |
| 39 | Seleccionar el boton siguiente | Se presenta la pantalla de editar representante legal |
| 40 | Seleccionar el boton siguiente | Se presenta la pantalla de editar info notaria |
| 41 | Seleccionar el boton guardar | Se presenta la pantalla de editar validacion documental |
| | | Se presenta el modal con el mensaje de actualizacion de datos |

Ilustración 74. DPC Editar Solicitud de Registro de Consultor. Elaboración Propia.

- Consulta Solicitud de Registro de Consultor

| 1 Escenario 7: Consultar solicitud | |
|--|--|
| 2 Descripción: Validar que una solicitud se pueda consultar. | |
| 3 | Clase: consultarConsultoresEsc01 |
| 4 | Metodo: consultarSolicitudConsultorEsc01 |
| 5 | Pasos a realizar |
| 6 | Resultados esperados |
| 7 | Ingresar al aplicativo de la gestora |
| 8 | Seleccionar el menu Personas |
| 9 | Seleccionar el submenu Gestionar |
| 10 | Seleccionar Consultor |
| 11 | En la grilla seleccionar el boton de consultar |
| 12 | Seleccionar el stepper de datos contratante |
| 13 | Dar click en ubicación contratante |
| 14 | Seleccionar el stepper representante legal |
| 15 | Dar click en ubicación representante legal |

Ilustración 75. DPC Escenario Uno Consulta Solicitud de Registro de Consultor. Elaboración Propia.

| 1 Escenario 8: Consultar solicitud | |
|---|--|
| 2 Descripción: Validar que una solicitud se pueda consultar en el proceso del stepper consultor | |
| 3 | Clase: consultarConsultoresEsc02 |
| 4 | Metodo: consultarSolicitudConsultorEsc01 |
| 5 | Pasos a realizar |
| 6 | Resultados esperados |
| 7 | Ingresar al aplicativo de la gestora |
| 8 | Seleccionar el menu Personas |
| 9 | Seleccionar el submenu Gestionar |
| 10 | Seleccionar Consultor |
| 11 | En la grilla seleccionar el boton de consultar |
| 12 | Seleccionar el stepper de consultor |
| 13 | Dar click en consultar |
| 14 | Dar click en aceptar |

Ilustración 76. DPC Escenario Dos Consulta Solicitud de Registro de Consultor. Elaboración Propia.

Dado que ya se tiene los diseños de pruebas de calidad (DPC) se inicia con la generación de los scripts, que serán requeridos dentro del proceso de pruebas automáticas, se generan todos los datos necesarios.

```

1 Insert into TPER_SOLICITUD_CONSULTOR (SCOID, SCOIDSOLICITUDPRINCIPAL, SCODSCANAL, SCODSTIPODECLARACION, SCOFFEFECHADECLARACION, SCODSESTAD
2 values ('-5', null, null, null, to_date('01/03/22', 'DD/MM/RR'), 'GRABADA', 'vUmhZzNcxq5E1bo', null, '6', '800046226', 'ADMINISTRADORA DE FONDOS DE PENSI
3 || TO_CLOB(q' [ora":null,"nombreRazonSocial":"ADMINISTRADORA DE FONDOS DE PENSIIONES", "matriculaComercio": "123", "sectorEconomico": "1", "actividadEc
4 || TO_CLOB(q' [":", "longitud": "", "correo": "123", "tipoCorreo": "INSTITUCIONAL", "idTipoCorreo": null, "celular": "123", "telefono": "", "extension": ""], "da
5 || TO_CLOB(q' [lidoCasada": null, "generoRepLegal": null, "indicadorCreacionRepLegal": "MANUAL", "tipoPersonaRepLegalEnum": "NATURAL", "idParametrizacion
6 || TO_CLOB(q' [erencia": null, "provincia": null, "departamento": null, "latitud": null, "longitud": null, "correo": null, "tipoCorreo": null, "idTipoCorreo": n
7 || TO_CLOB(q' [44, "fechaInicioAdeunda": "2022-02-01", "fechaFinAdeunda": "2022-02-20", "numeroIdentificacion": "800227940", "idTipoIdentificacion": 20, "
8 COMMIT
9

```

Ilustración 77. Script Registro Consultor. Elaboración Propia.

```

1  DELETE FROM TPER_SOLICITUD_CONSULTOR WHERE SCOID IN (-5);
2  COMMIT
3

```

Ilustración 78. Delete Script Registro Consultor. Elaboración Propia.

```

1  Insert into TPER_SOLICITUD_CONSULTOR (SCOID, SCOIDSOLICITUDPRINCIPAL, SCODSCANAL, SCODSTIPODECLARACION, SCOFEFECHADECLARACION, SCODSEST.
2  values ('-6', null, null, null, to_date('04/02/22', 'DD/MM/RR'), 'GRABADA', 'n00m2vr4hZyRak1', null, '6', '800046226', 'PRUEBA', null, null, null, '{
3  "idSolicitudConsultor":1,
4  "datosRepresentanteLegal":{
5  "idTipoIdentificacionRepLegal":6,
6  "numeroIdentificacionRepLegal":"123123",
7  "complemento":"567",
8  "codigoTipoIdentificacionRepLegal":"NIT",
9  "primerNombreRepLegal":"DANIEL",
10 "segundoNombreRepLegal":"FELIPE",
11 "primerApellidoRepLegal":"RUEDA",
12 "segundoApellidoRepLegal":"MARINO",
13 "apellidoCasada":"ABC",
14 "generoRepLegal":"FEMENINO"
15 },
16 "datosSolicitud":{
17 "numDeclaracion":"hhy6",
18 "canal":"AUTOMATICO",
19 "documentosPresentados":"IMPRESO",
20 "constituyeAgenteRetencion":1,
21 "nombreGestor":"JALZATE",
22 "lugarDeclaracion":"Bolivia",
23 "estado":"GRABADA",
24 "observaciones":"NN"
25 },

```

Ilustración 79. Script Registro Consultor. Elaboración Propia.

```

1  DELETE FROM TPER_SOLICITUD_CONSULTOR WHERE SCOID IN (-6);
2  COMMIT
3

```

Ilustración 80. Script Registro Consultor. Elaboración Propia.

Se inician las pruebas automáticas, se dividieron de acuerdo con las funcionalidades a probar de registro de consultores:

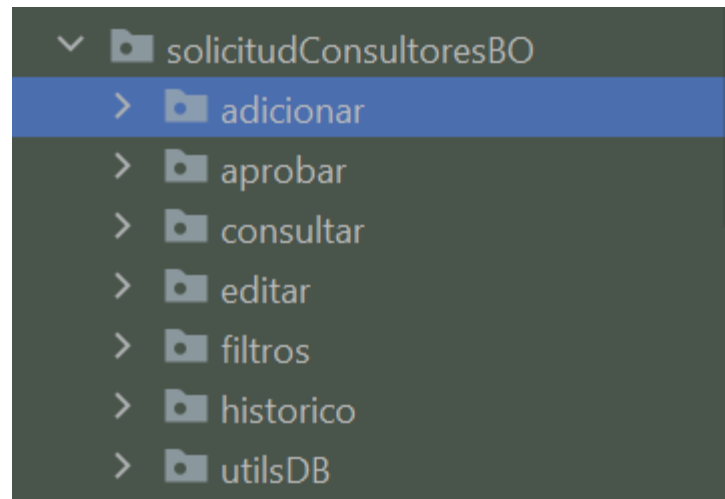


Ilustración 81. Carpeta Funcionalidad Solicitud Consultor. Elaboración Propia.

Se pondré un ejemplo de las pruebas de adicionar solicitud de registro de consultor a continuación en la ilustración 82:

Clase dto

```

3  /**
4  * <b> Descripción:<b> Clase que determina los campos de creación de adicionar consultor
5  * <b>Caso de uso:<b>
6  *
7  * @author Danna Michelle Arévalo Henández <darevalo@heinsohn.com.co>
8  * @version
9  *
10
11 public class AdicionarConsultorDto {
12
13     private String canal;
14     private String numeroDeclaracion;
15     private String fechaDeclaracion;
16     private String documentosPresentados;
17     private Boolean agenteRetencion;
18     private String gestor;
19
20
21
22     /**
23     * @return the canal
24     */
25     public String getCanal() { return canal; }
26
27
28
29     /**

```

Ilustración 82. Clase Dto Adicionar Registro Consultor. Elaboración Propia.

```

29     /**
30      * @param canal the canal to set
31      */
32     public void setCanal(String canal) { this.canal = canal; }
33
34
35
36     /**
37      * @return the numeroDeclaracion
38      */
39     public String getNumeroDeclaracion() { return numeroDeclaracion; }
40
41
42
43     /**
44      * @param numeroDeclaracion the numeroDeclaracion to set
45      */
46     public void setNumeroDeclaracion(String numeroDeclaracion) { this.numeroDeclaracion = numeroDeclaracion; }
47
48
49
50     /**
51      * @return the fechaDeclaracion
52      */
53     public String getFechaDeclaracion() { return fechaDeclaracion; }
54
55
56
57     /**
58      * @param fechaDeclaracion the fechaDeclaracion to set
59      */
60     public void setFechaDeclaracion(String fechaDeclaracion) { this.fechaDeclaracion = fechaDeclaracion; }
61
62
63

```

Ilustración 83. Clase Dto Adicionar Registro Consultor. Elaboración Propia.

Page

```

8     /**
9      * <b> Descripción:</b> Clase que contiene las acciones para crear un consultor
10     * <b> Caso de uso:</b>
11     *
12     * @author Danna Michelle Arévalo Henández <darevalo@heinsohn.com.co>
13     * @version
14     *
15     */
16     public class AdicionarConsultorPage {
17
18         private By canal = By.id("IDcanal");
19         private By numeroDeclaracion = By.id("numeroSolicitudId");
20         private By fechaDeclaracion = By.id("fechaSolicitudInput");
21         private By documentosPresentados = By.id("documentosPresentadosId");
22         private By gestor = By.id("gestorId");
23         private By constituyeRetencionNo = By.id("constituyeAgenteRetencionNo");
24         private By constituyeRetencionSi = By.id("constituyeAgenteRetencionSi");
25         private By btnSiguiente = By.id("wzSiguienteId");
26
27         private By msnModal = By.id("idMensajeModal");
28         private By btnContinuar = By.id("btnContinuar");
29         private By cardConsultor = By.id("consultor");
30         private By btnAdicionar = By.id("btnAdicionar");
31
32         private final WebDriver driver;
33         private final SeleniumWebDriverUtils utils;

```

Ilustración 84. Clase Page Adicionar Registro Consultor. Elaboración Propia.

```

32     private final WebDriver driver;
33     private final SeleniumWebDriverUtils utils;
34
35     public AdicionarConsultorPage(WebDriver driver) {
36         this.driver = driver;
37         utils = new SeleniumWebDriverUtils(this.driver);
38     }
39
40     /**
41      *
42      * Método encargado de ingresar adicionar los datos de primer steper de consultor
43      *
44      * @author Danna Michelle Arevalo Hernandez <darevalo@heinsohn.com.co>
45      *
46      *
47      */
48     @ public void adicionarConsultor (AdicionarConsultorDto adicionarConsultorDto) {
49
50         utils.waitForOneSeconds();
51         utils.waitForOneSeconds();
52
53         utils.click(btnAdicionar);
54
55         utils.waitForOneSeconds();
56         utils.waitForOneSeconds();
57         utils.waitForOneSeconds();

```

Ilustración 85. Método Clase Page Adicionar Registro Consultor. Elaboración Propia.

```

60
61     // Seleccionamos el canal para el empleador
62     if (adicionarConsultorDto.getCanal() != null) {
63         utils.selectOptionList(canal, adicionarConsultorDto.getCanal());
64     }
65
66     utils.waitForOneSeconds();
67     utils.waitForOneSeconds();
68
69     // Le asignamos un numero de declaracion al empleador
70     if (adicionarConsultorDto.getNumeroDeclaracion() != null) {
71         utils.type(numeroDeclaracion, adicionarConsultorDto.getNumeroDeclaracion());
72     }
73
74     utils.waitForOneSeconds();
75     utils.waitForOneSeconds();
76
77     // Seleccionamos la fecha de declaracion
78     if (adicionarConsultorDto.getFechaDeclaracion() != null) {
79         utils.type(fechaDeclaracion, adicionarConsultorDto.getFechaDeclaracion());
80     }
81
82     utils.waitForOneSeconds();
83     utils.waitForOneSeconds();
84

```

Ilustración 86. Método Clase Page Adicionar Registro Consultor. Elaboración Propia.

Test

```

1 package com.heinsohn.personas.solicitudConsultoresB0.adicionar.test;
2
3 import ...
20
21
22 * <b> Descripción:<b> Clase que permite ejecutar el test de creación de solicitud de consultor
23 * <b> Caso de uso:<b> CUPE1418
24 *
25 * @author Danna Michelle Arevalo Hernandez <darevalo@heinsohn.com.co>
26 * @version
27 *
28 */
29 public class AdicionarConsultorTest02 {
30
31     private SeleniumTest selenium;
32     private LoginPage LoginPage;
33     private MenuPage menuPage;
34     private AdicionarConsultorPage adicionarConsultorPage;
35     private AdicionarDatosContratantePage adicionarDatosContratantePage;
36     private AdicionarRepresentanteLegalPage adicionarRepresentanteLegalPage;
37     private EntrarCardPersonasUtils entrarCardPersonasUtils;
38
39     List<String> filaEncontrada;
40

```

Ilustración 87. Clase Test Adicionar Consultor. Elaboración Propia.

```

41 /**
42 *
43 * Metodo encargado de implementar las instancias antes de la ejecucion de la prueba
44 *
45 * @author Jhonatan Morales <jhmorales@heinsohn.com.co>
46 *
47 */
48 @BeforeMethod(groups = { "consultorGestora" })
49 public void iniciarPrueba() {
50     // Se evalua que el objeto selenium no tenga instancias previas
51     if (selenium == null) {
52         selenium = new SeleniumTest(getClass().getSimpleName());
53         loginPage = new LoginPage(selenium.getDriver());
54         menuPage = new MenuPage(selenium.getDriver());
55         adicionarConsultorPage = new AdicionarConsultorPage(selenium.getDriver());
56         adicionarDatosContratantePage = new AdicionarDatosContratantePage(selenium.getDriver());
57         adicionarRepresentanteLegalPage = new AdicionarRepresentanteLegalPage(selenium.getDriver());
58         entrarCardPersonasUtils = new EntrarCardPersonasUtils(selenium.getDriver());
59         filaEncontrada = new ArrayList<>();
60     }
61 }
62
63
64 }

```

Ilustración 88. Clase Test Adicionar Consultor. Elaboración Propia.

```

66  /**
67  * Metodo encargado de crear una solicitud de consultor
68  *
69  * @author Danna Michelle Arevalo Hernandez <darevalo@heinsohn.com.co>
70  *
71  */
72  @Test(groups = { "consultorGestora" })
73  public void AdicionarConsultorTest02() {
74
75      // Ingresar a Personas
76
77      LoginPage.login(SeleniumProperties.getPropiedad(ConfigSeleniumPropiedades.USUARIO),
78                    SeleniumProperties.getPropiedad(ConfigSeleniumPropiedades.CONTRASENA));
79
80      selenium.getUtils().waitForLoading(Integer.parseInt(SeleniumProperties.getPropiedad(ConfigSeleniumPropiedades.TIEMPO_CARGA)));
81      selenium.getUtils().waitForLoading(Integer.parseInt(SeleniumProperties.getPropiedad(ConfigSeleniumPropiedades.TIEMPO_CARGA)));
82
83      selenium.getUtils().waitForLoading(Integer.parseInt(SeleniumProperties.getPropiedad(ConfigSeleniumPropiedades.TIEMPO_CARGA)));
84      selenium.getUtils().waitForLoading(Integer.parseInt(SeleniumProperties.getPropiedad(ConfigSeleniumPropiedades.TIEMPO_CARGA)));
85
86      menuPage.clickOnMenu(menuPage.personas, menuPage.gestionar);
87
88      // En el modulo de gestionar entramos a la card de personas
89      entrarCardPersonasUtils.entrarCardSolicitudes();
90

```

Ilustración 89. Método Clase Test Adicionar Consultor. Elaboración Propia.

```

91      // Ingresamos a la card de consultores
92      adicionarConsultorPage.clickCardConsultores();
93
94      //Le damos click en adicionar consultor
95      AdicionarConsultorDto adicionarConsultorDto = new AdicionarConsultorDto();
96      adicionarConsultorDto.setCanal("Automático");
97      adicionarConsultorDto.setFechaDeclaracion("03/02/2022");
98      adicionarConsultorDto.setDocumentosPresentados("Solo formulario");
99      adicionarConsultorDto.setAgenteRetencion(Boolean.TRUE);
100     adicionarConsultorDto.setGestor("312-MARISOL BAUTISTA");
101
102     adicionarConsultorPage.adicionarConsultor(adicionarConsultorDto);
103
104     // Adicionamos datos contratante
105     AdicionarDatosContratanteDto adicionarDatosContratanteDto = new AdicionarDatosContratanteDto();
106     adicionarDatosContratanteDto.setTipoDeIdentificacion("60B");
107     adicionarDatosContratanteDto.setNumeroDeIdentificacion("01022573028");
108     adicionarDatosContratanteDto.setDireccionAdministrativa("00090");
109     adicionarDatosContratanteDto.setCodigoUnidadEjecutora("1");
110     adicionarDatosContratanteDto.setCiudad("COCHABAMBA / SECCION CAPITAL / CERCADO");
111     adicionarDatosContratanteDto.setCorreoElectronico("agimex@bolivia.com.bo");
112     adicionarDatosContratanteDto.setTipoDeCorreo("Laboral");
113     adicionarDatosContratanteDto.setCelular("3200667722");
114     adicionarDatosContratanteDto.setTelefono("3759000");
115     adicionarDatosContratanteDto.setExtension("2");

```

Ilustración 90. Método Clase Test Adicionar Consultor. Elaboración Propia.

```

118
119     adicionarDatosContratantePage.ingresarDatosContratante(adicionarDatosContratanteDto);
120
121     //Se ingresan todos los campos de representante legal
122     AdicionarRepresentanteLegalConsultorDto adicionarRepresentanteLegalConsultorDto = new AdicionarRepresent
123     adicionarRepresentanteLegalConsultorDto.setTipoIdRepresentLegal("Carnet de extranjerero");
124     adicionarRepresentanteLegalConsultorDto.setNumeroIdRepresentLegal("9000235583");
125     adicionarRepresentanteLegalConsultorDto.setGenero("Femenino");
126     adicionarRepresentanteLegalConsultorDto.setCiudadRepresentLegal("Cochabamba");
127     adicionarRepresentanteLegalConsultorDto.setCorreoElectronicoRepresentLegal("ricardo@bolivia.com.bo");
128     adicionarRepresentanteLegalConsultorDto.setTipoDeCorreoRepresentLegal("Personal");
129     adicionarRepresentanteLegalConsultorDto.setCelularRepresentLegal("3200667345");
130
131     adicionarRepresentanteLegalPage.ingresarRepresentanteLegal(adicionarRepresentanteLegalConsultorDto);
132
133     //Cerramos la sesion
134     loginPage.logout();
135 }
136
137 @AfterMethod(groups = { "consultorGestora" })
138 public void cerrar() {
139     selenium.getDriver().close();
140 }
141 }
142

```

Ilustración 91. AfterMethod Clase Test Adicionar Consultor. Elaboración Propia.

Considerando que se tiene realizaron todas las pruebas automáticas correspondientes a los escenarios de los diseños de casos de uso de prueba, se procede a hacer la ejecución de dichas pruebas con el siguiente comando de la ilustración 92.

```

C:\Windows\System32\cmd.exe
Microsoft Windows [Versión 10.0.19043.1526]
(c) Microsoft Corporation. Todos los derechos reservados.

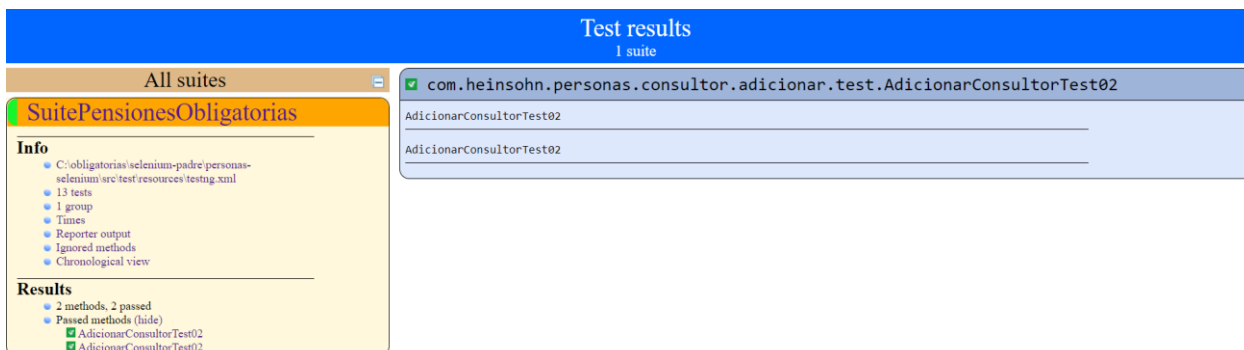
C:\obligatorias\selenium-padre\personas-selenium>mvn test -fae -P selenium -Dgroups=consultorGestora -Dmaven.test.skip=false_

```

Ilustración 92. Comando ejecución de pruebas automáticas en grupo. Elaboración Propia.

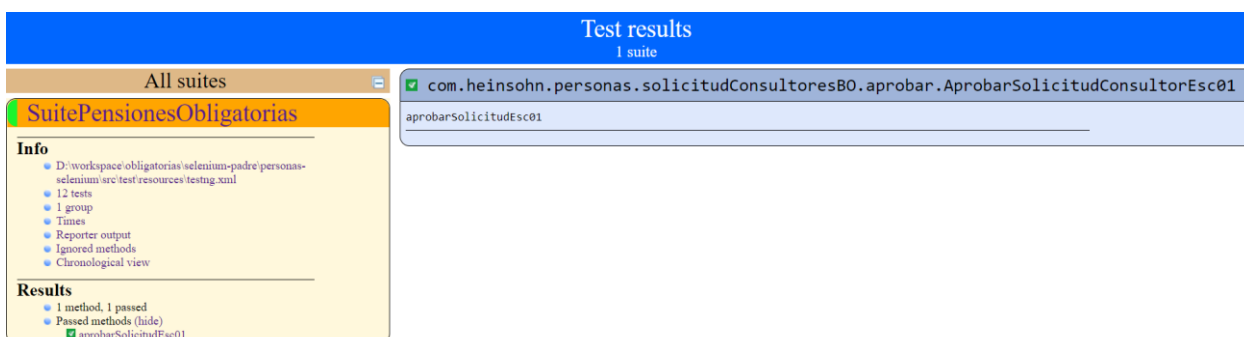
Cuando se ejecuta dicho comando se correrán automáticamente todas las pruebas que estén dentro del grupo de pruebas que se esté lanzando, como se mencionó anteriormente en la ruta donde se tiene las pruebas, en la carpeta de target > surefire-reports > personasInfoBasica > index.html se encontraran todos los reportes de las pruebas que se ejecutaron automáticamente en ese grupo.

Reportes Pruebas Solicitud de Registro Consultor



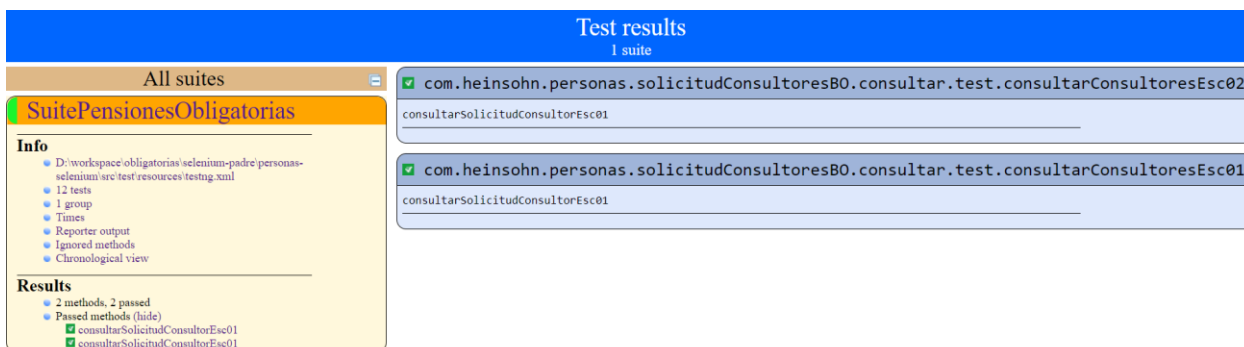
The screenshot shows a test results interface. On the left, under 'All suites', the 'SuitePensionesObligatorias' is expanded, showing 'Info' (13 tests, 1 group, Times, Reporter output, Ignored methods, Chronological view) and 'Results' (2 methods, 2 passed, Passed methods (hide), AdicionarConsultorTest02, AdicionarConsultorTest02). The main area shows 'Test results' for '1 suite' with a green checkmark for 'com.heinsohn.personas.consultor.adicionar.test.AdicionarConsultorTest02'. Below this, two test cases are listed: 'AdicionarConsultorTest02' and another 'AdicionarConsultorTest02', both with green checkmarks.

Ilustración 93. Reportes Pruebas Adicionar Solicitud de Consultor. Elaboración Propia.



The screenshot shows a test results interface. On the left, under 'All suites', the 'SuitePensionesObligatorias' is expanded, showing 'Info' (12 tests, 1 group, Times, Reporter output, Ignored methods, Chronological view) and 'Results' (1 method, 1 passed, Passed methods (hide), aprobarSolicitudEsc01). The main area shows 'Test results' for '1 suite' with a green checkmark for 'com.heinsohn.personas.solicitudConsultoresB0.aprobar.AprobarSolicitudConsultorEsc01'. Below this, one test case is listed: 'aprobarSolicitudEsc01' with a green checkmark.

Ilustración 94. Reportes Pruebas Aprobar Solicitud de Consultor. Elaboración Propia.



The screenshot shows a test results interface. On the left, under 'All suites', the 'SuitePensionesObligatorias' is expanded, showing 'Info' (12 tests, 1 group, Times, Reporter output, Ignored methods, Chronological view) and 'Results' (2 methods, 2 passed, Passed methods (hide), consultarSolicitudConsultorEsc01, consultarSolicitudConsultorEsc01). The main area shows 'Test results' for '1 suite' with green checkmarks for 'com.heinsohn.personas.solicitudConsultoresB0.consultar.test.consultarConsultoresEsc02' and 'com.heinsohn.personas.solicitudConsultoresB0.consultar.test.consultarConsultoresEsc01'. Below these, two test cases are listed: 'consultarSolicitudConsultorEsc01' and another 'consultarSolicitudConsultorEsc01', both with green checkmarks.

Ilustración 95. Reportes Pruebas Consultar Solicitud de Consultor. Elaboración Propia.

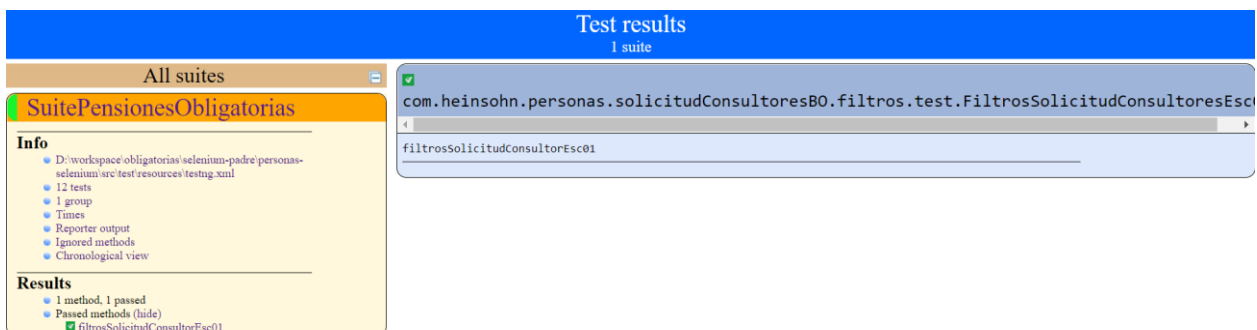


Ilustración 96. Reportes Pruebas Filtros Solicitud de Consultor. Elaboración Propia.

6.4. Funcionalidad Desclasificación de asegurados militares

La funcionalidad de desclasificación de asegurados militares es el proceso en el cual un militar empieza la solicitud de desclasificación de militar a asegurado.

Lo primero que se hizo fue las pruebas de interfaz y verificación de id's de los elementos de las pantallas de la funcionalidad de Desclasificación de Asegurados Militares como vemos en la ilustración 97.

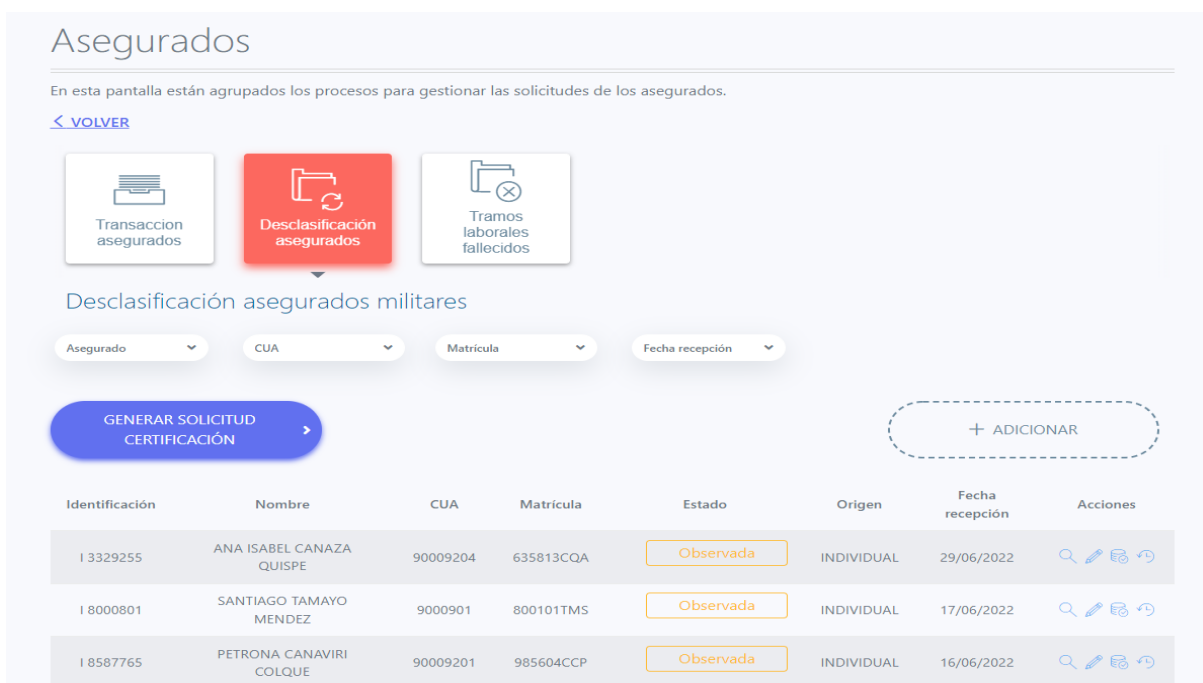


Ilustración 97. Funcionalidad Desclasificación Asegurados Militares. Elaboración Propia.

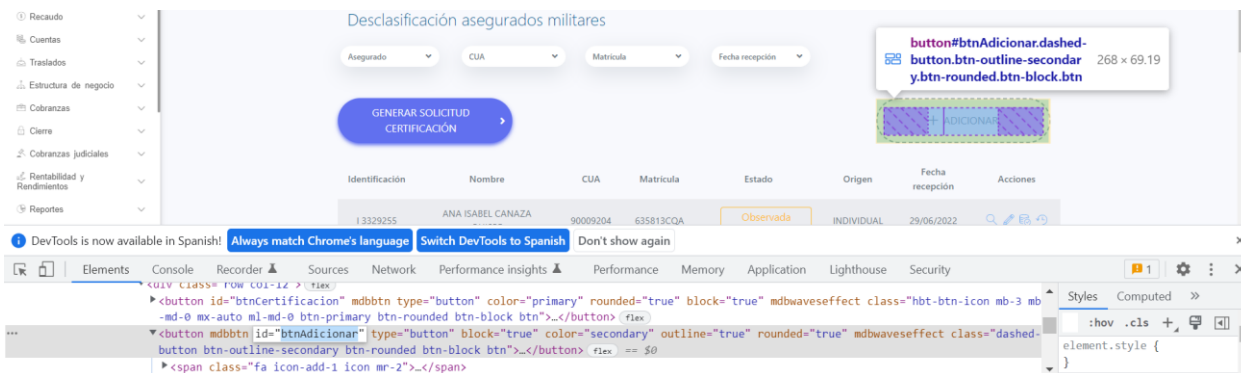


Ilustración 98. Verificación Localizadores id's. Elaboración Propia.

Para poder realizar el diseño de los DPC lo primero que se hizo fue la lectura de los casos de uso de desclasificación de asegurados militares que se muestran a continuación:

- CUPE1436 BOL Crear-Modificar solicitud de desclasificación de militar
- CUPE1437 BOL Cargar solicitudes de desclasificación de militares
- CUPE1434 BOL Gestionar solicitudes de desclasificación de militares
- CUPE1435 BOL Ingresar-Modificar solicitud de desclasificación de militar

Una vez se que se hace la lectura de los casos de uso y se sabe los requerimientos que pide el cliente empezamos con el diseño de pruebas de calidad, creando así diversos escenarios para la funcionalidad de desclasificación de asegurados militares, los DPC que se realizaron se muestran a continuación:

- Adicionar Solicitud de Desclasificación de Asegurados Militares

| HEINSOHN BUSINESS TECHNOLOGY | | PTL-GINF-003 Versión formato: 2016/04/01 Desarrollado por Heinsohn Business Technology |
|------------------------------|---|--|
| DOCUMENTO PLAN DE PRUEBAS | | |
| 1 | | |
| 2 | | |
| 3 | | |
| 4 | Fecha_ | 25/02/2022 |
| 5 | Título de la prueba / Escenario | Crear Solicitud de Desclasificación de asegurado militar |
| 6 | | 1. CUPEI436 Crear solicitud de desclasificación de militar |
| 7 | | 2. |
| 8 | Casos de uso | 3. |
| 9 | | 4. |
| 10 | | 5. |
| 11 | Clase / Metodo | Clase: AdicionarSolicitudDesclaAseguradosTest01 Metodo: adicionarSolicitudDesclaAseguradosTest01 |
| 12 | Usuario que ejecuta la prueba | Danna Michelle Arévalo Hernández |
| 13 | | |
| 14 | Escenario 1 Crear Solicitud de desclasificación | |
| 15 | Descripción: Permite crear una solicitud y que quede en estado aprobada | |
| 16 | Pasos a realizar | Resultados esperados |
| 17 | Ejecuta en el navegador la URL: https://pensiones-ci.heinsohn.com.co:8081/#/ | Presenta la pantalla "Pensiones / Cesantías / Inversion" con los campos Usuario, Clave y Ingresar |
| 18 | Ingrese los datos de entrada Usuario: selenium, Clave: selenium, haga clic en el botón "INGRESAR" | Presenta el menú que corresponde a la aplicación de Pensiones Obligatorias |
| 19 | Seleccione el menú "Personas" | Despliega el submenú "Gestionar, Vision 360 Afiliado, RAE" |
| 20 | Seleccionar el submenú Gestionar | Se presenta las card de "Personas" y "Solicitudes" |
| 21 | Seleccionamos la card de Asegurados | Se presenta la pantalla con las solicitudes de asegurados |
| 22 | Se le da click en el boton de "Adicionar" | Se presenta la pantalla de Adicionar Asegurados en el stepper de solicitud |
| 23 | Ingresamos los siguientes datos: - Cite Ministerio: 91 - Fecha de Cite: 28/02/2022 - Fecha de recepción: 22/02/2022 - Cua: 567 - Gestor: 312-MARISOL BAUTISTA | Se presenta la stepper de Datos de Asegurado |
| 24 | Le damos click en el boton de siguiente Ingresamos los siguientes datos: - Tipo De Identificación: Carnet de extranjero (E) - Numero De Identificación: 4000500 - Fecha Nacimiento: 20/01/1969 - Primer Nombre: RODRIGO - SegundoNombre: ANDRES - PrimerApellido: GOMEZ - SegundoApellido: AMEZQUITA - Nacionalidad: BOLIVIA - PaisDeNacimiento: BOLIVIA - Ciudad: SUCRE / OROPEZA / CHUQUISACA - Genero: Masculino - EstadoCivil: SOLTERO(A) - Matricula: 690120GMR - CiudadUbicacion: COCHABAMBA / CERCADO / COCHABAMBA - Zona: Queru queru - Direccion: Av. Americas - Numero: 500 - Referencia: Pruebas - CorreoElectronico: pruebas@gestora.com.bo - TipoDeCorreoElectronico: Personal - Celular: 3202272833 | Se presenta la stepper de Validación documental |
| 25 | Le damos click en el boton de siguiente Ingresamos el documento Nit solicitado | Se presenta el mensaje de éxito de guardado de documentos "Has adicionado exitosamente la solicitud de desclasificación de asegurado militar." |
| 26 | Le damos click en el boton de guardar | |
| 27 | Verificamos que haya quedado el registro guardado | |
| 28 | Ingresamos el documento Nit solicitado | Se presenta el mensaje de éxito de guardado de documentos "Has adicionado exitosamente la solicitud de desclasificación de asegurado militar." |
| 29 | Le damos click en el boton de validar | |
| 30 | Le damos click en el boton de guardar | Se presenta el mensaje de éxito de guardad de solicitud "Has adicionado exitosamente la solicitud de desclasificación de asegurado militar." |
| 31 | Verificamos que haya quedado el registro guardado | |

Ilustración 99. DPC Adicionar Solicitud de Desclasificación de Asegurados Miliars. Elaboración Propia.

- Consultar Solicitud de Desclasificación de Asegurados Militares

| | | | |
|----|--|---|--|
| 1 | | HEINSOHN BUSINESS TECHNOLOGY | PTL-GINF-003 Versión formato: 2016/04/01 Desarrollado por Heinsohn Business Technology |
| 2 | | DOCUMENTO PLAN DE PRUEBAS | |
| 3 | | | |
| 4 | Fecha_ | 25/02/2022 | |
| 5 | Título de la prueba / Escenario | Consultar Solicitud de Desclasificación de asegurado militar | |
| 6 | | 1. | CUPE1434 Gestionar solicitudes de desclasificación de militares |
| 7 | | 2. | |
| 8 | Casos de uso | 3. | |
| 9 | | 4. | |
| 10 | | 5. | |
| 11 | Clase / Metodo | Clase: ConsultarDesclaAseguradosTest01 | Metodo: consultarDesclaAseguradosTes |
| 12 | Usuario que ejecuta la prueba | Danna Michelle Arévalo Hernández | |
| 13 | | | |
| 14 | Escenario 1: Consultar Solicitud de desclasificación | | |
| 15 | <i>Descripción: Permite consultar una solicitud</i> | | |
| 16 | Pasos a realizar | Resultados esperados | |
| 17 | Ejecuta en el navegador la URL: https://pensiones-cl.heinsohn.com.co:8081/#/ | Presenta la pantalla "Pensiones / Cesantías / Inversión" con los campos Usuario, Clave y Ingresar | |
| 18 | Ingrese los datos de entrada Usuario: selenium, Clave: selenium, haga clic en el botón "INGRESAR" | Presenta el menú que corresponde a la aplicación de Pensiones Obligatorias | |
| 19 | Seleccione el menú "Personas" | Despliega el submenú "Gestionar, Vision 360 Afiliado, RAE" | |
| 20 | Seleccionar el submenú Gestionar | Se presenta las card de "Personas", "Solicitudes", "Asegurados" | |
| 21 | Seleccionamos la card de Asegurados | Se presenta la pantalla de consulta de solicitud | |
| 22 | Se le da click en el boton de consultar | Se presenta la pantalla de Adicionar Asegurados en el steper de solicitud | |
| | Verificamos los siguientes datos: - Cite Ministerio: 91 - Fecha de Cite: 28/02/2022 - Fecha de recepción: 22/02/2022 - Cua: 567 - Gestor: MARISOL BAUTISTA - Estado: Observada - Observaciones: Pendiente documentación Matrícula diferente | Se presenta la card de Datos de Asegurado | |
| 23 | Le damos click en la card de datos de asegurado | | |
| 24 | Verificamos los siguientes datos: - Tipo De Identificación: Carnet de extranjero (E) - Numero De Identificación: 4000504 - Nombre Asegurado: RODRIGO GOMEZ - Fecha Nacimiento: 10/02/1993 - Nacionalidad: BOL - PaisDeNacimiento: BOLIVIA - Ciudad: CHAUNACA - Provincia: OROPEZA - Departamento: CHUQUISACA - Genero: MASCULINO - EstadoCivil: SOLTERO(A) - Matrícula: 690120GMR - CiudadUbicación: COCHABAMBA / CERCADO / COCHABAMBA - Zona: Queru queru - Dirección: Av. Americas - Numero: 500 - Referencia: Pruebas - CorreoElectronico: pruebas@gestora.com.bo - TipoDeCorreoElectronico: Personal - Celular: 3202272833 Le damos click en la card de validación documental | Se presenta la card de Validación documental | |
| 25 | Verificamos el documento Nit Cargado | Se presenta la imagen o pdf | |

Ilustración 100. DPC Consultar Solicitud de Desclasificación de Asegurados Militares. Elaboración Propia.

- Filtros de Solicitud de Desclasificación de Asegurados Militares

| | | | |
|----|--|---|--|
| 1 | | HEINSOHN BUSINESS TECHNOLOGY | PTL-GINF-003 Versión formato: 2016/04/01 Desarrollado por Heinsohn Business Technology |
| 2 | | DOCUMENTO PLAN DE PRUEBAS | |
| 3 | | | |
| 4 | Fecha_ | 25/02/2022 | |
| 5 | Título de la prueba / Escenario | Filtros de Solicitud de Desclasificación de asegurado militar | |
| 6 | | 1. | CUPE1434 Gestionar solicitudes de desclasificación de militares |
| 7 | | 2. | |
| 8 | Casos de uso | 3. | |
| 9 | | 4. | |
| 10 | | 5. | |
| 11 | Clase / Metodo | Clase: FiltrarAseguradoTest01 | Metodo: filtrarAseguradoTest01 |
| 12 | Usuario que ejecuta la prueba | Danna Michelle Arévalo Hernández | |
| 13 | | | |
| 14 | Escenario 1: Filtrar Solicitud de Desclasificación por Asegurado | | |
| 15 | <i>Descripción: Permite filtrar una solicitud por asegurado</i> | | |
| 16 | Pasos a realizar | Resultados esperados | |
| 17 | Ejecuta en el navegador la URL: https://pensiones-ci.heinsohn.com.co:8081/#/ | Presenta la pantalla "Pensiones / Cesantías / Inversión" con los campos Usuario, Clave y Ingresar | |
| 18 | Ingrese los datos de entrada Usuario: selenium, Clave: selenium, haga clic en el botón "INGRESAR" | Presenta el menú que corresponde a la aplicación de Pensiones Obligatorias | |
| 19 | Seleccione el menú "Personas" | Despliega el submenú "Gestionar, Vision 360 Afiliado, RAE" | |
| 20 | Seleccionar el submenú Gestionar | Se presenta las card de "Personas", "Solicitudes", "Asegurados" | |
| 21 | Seleccionamos la card de Asegurados | Se presenta la pantalla de consulta de solicitud | |
| 22 | Se le da click en el filtro de Asegurado | Se presentan los campos del filtro | |
| 23 | Ingresamos los datos del filtro: - Tipo de identificación asegurado: Carnet de extranjero (E) - Numero de identificación asegurado: 4000400 | Se presenta la solicitud del asegurado que filtramos | |
| 24 | Le damos click en boton de aplicar filtro | | |
| | Verificamos los campos que trajo la solicitud del asegurado: - TipoDeIdentificacion: E 4000400 - Nombre: MARIA CAMILA CASTRO BENAVIDEZ - Cua: 456 - Matricula: 965510CEM - Estado: Aprobada - Origen: INDIVIDUAL - FechaDeRecepcion: 2022-02-25 | Cerramos sesión | |

Ilustración 101. DPC Filtros Desclasificación de Asegurados Militares. Elaboración Propia.

- Generación de Certificación

| | | | |
|----|--|--|---|
| 2 | | DOCUMENTO PLAN DE PRUEBAS | Version formato: 2016/04/01 Desarrollado por Heinsohn Business Technology |
| 3 | | | |
| 4 | Fecha | 25/02/2022 | |
| 5 | Título de la prueba / Escenario | Generar Certificación | |
| 6 | | 1. | CUPE1434 Gestionar solicitudes de desclasificación de militares |
| 7 | | 2. | |
| 8 | Casos de uso | 3. | |
| 9 | | 4. | |
| 10 | | 5. | |
| 11 | Clase / Metodo | Clase: GenerarCertificacionTest01 | Metodo: generarCertificacionTest01 |
| 12 | Usuario que ejecuta la prueba | Danna Michelle Arévalo Hernández | |
| 13 | | | |
| 14 | Escenario 1: Generar certificación asegurado | | |
| 15 | <i>Descripción: Permite generar certificación a un asegurado militar</i> | | |
| 16 | Pasos a realizar | Resultados esperados | |
| 17 | Ejecuta en el navegador la URL: https://pensiones- ci.heinsohn.com.co:8081/#/ | Presenta la pantalla "Pensiones / Cesantías / Inversión" con los campos Usuario, Clave y Ingresar | |
| 18 | Ingrese los datos de entrada Usuario: selenium, Clave: selenium, haga clic en el botón "INGRESAR" | Presenta el menú que corresponde a la aplicación de Pensiones Obligatorias | |
| 19 | Seleccione el menú "Personas" | Despliega el submenú "Gestionar, Vision 360 Afiliado, RAE" | |
| 20 | Seleccionar el submenú Gestionar | Se presenta las card de "Personas", "Solicitudes", "Asegurados" | |
| 21 | Seleccionamos la card de Asegurados | Se presenta la pantalla de consulta de solicitud | |
| 22 | Se el boton de Generar Certificación | Se presentan todos los datos que se deben diligenciar | |
| 23 | Ingresamos los siguientes datos - Cua: 567 - Matricula: 690120GMR - CiteMinisterio: 91 - TipoDelIdentificacion: Carnet de extranjero (E) - NumeroDelIdentificacion: 4000503 - NombresYApellidos: RODRIGO GOMEZ - FechaNacimiento: 20/01/1969 Le damos click en el boton de generar certificacion | Se presenta el mensaje modal de guardado con éxito y se descarga automáticamente la carta de certificación | |
| 24 | Le damos click en el boton de Aceptar | Cerramos sesión | |
| 25 | | | |

Ilustración 102. DPC Generar Certificación de Militar. Elaboración Propia.

Ya que se tiene los DPC de los escenarios que se van a probar para esta funcionalidad, se procede a la creación de las carpetas y las clases como se evidencian en las ilustraciones 103, 104 y 105.

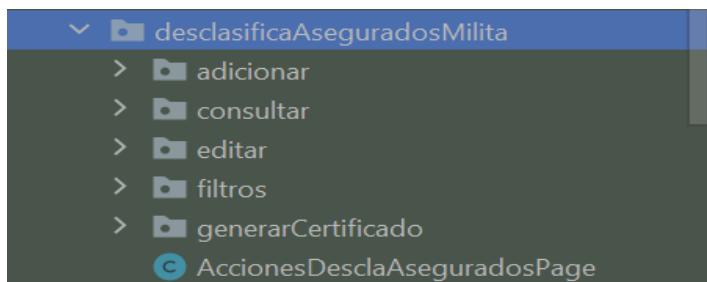


Ilustración 103. Carpetas Funcionalidad Desclasificación. Elaboración Propia.

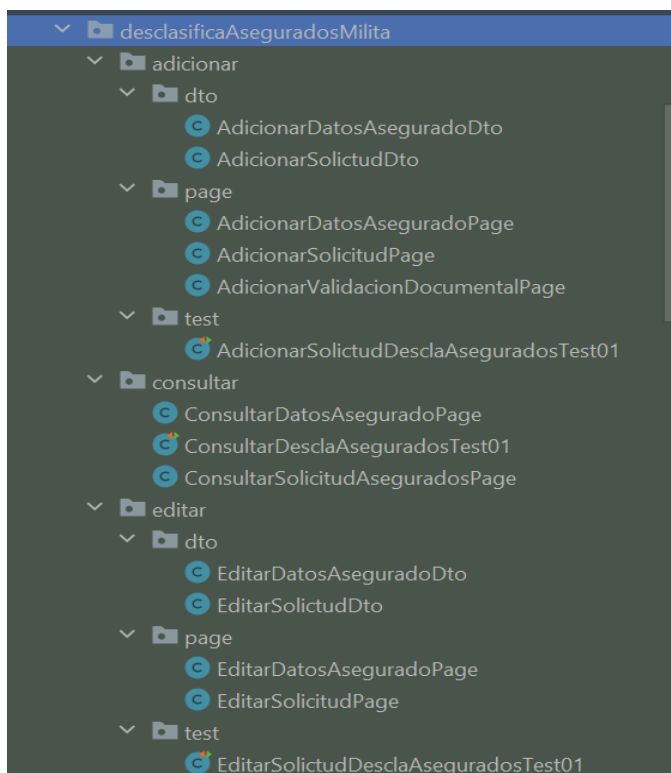


Ilustración 104. Carpetas Funcionalidad Desclasificación. Elaboración Propia.

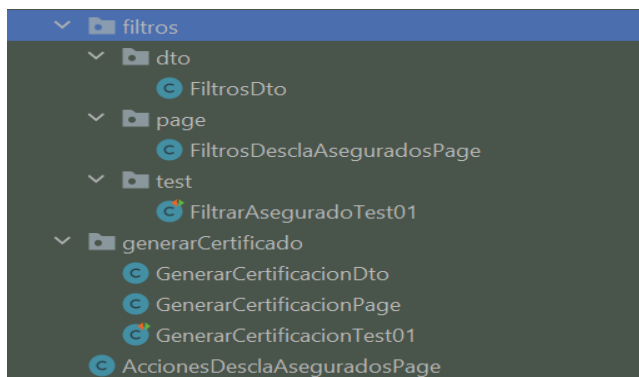


Ilustración 105. Carpetas Funcionalidad Desclasificación. Elaboración Propia.

Se procede a la elaboración de los test, page y dto de los escenarios que se crearon en el DPC, se mostrara un ejemplo de las pruebas que se realizaron.

Test

```

1 package com.heinsohn.personas.desclasificaAseguradosMilita.adicionar.test;
2
3 import ...
4
25
26 /**
27  * <b> Descripción:<b> Clase que permite ejecutar el test de desclasificación de asegurados militares
28  * <b> Caso de uso:<b> CUPE1436
29  *
30  * @author Danna Michelle Arevalo Hernandez <darevalo@heinsohn.com.co>
31  * @version
32  *
33  */
34 public class AdicionarSolicitudDesclaAseguradosTest01 {
35
36     private SeleniumTest selenium;
37     private LoginPage loginPage;
38     private MenuPage menuPage;
39     private AccionesDesclaAseguradosPage accionesDesclaAseguradosPage;
40     private AdicionarSolicitudPage adicionarSolicitudPage;
41     private AdicionarDatosAseguradoPage adicionarDatosAseguradoPage;
42     private EntrarCardPersonasUtils entrarCardPersonasUtils;
43
44     List<String> filaEncontrada;
45

```

Ilustración 106. Test Adicionar Solicitud de Desclasificación de Asegurado Militar. Elaboración Propia.

```

46 /**
47  *
48  * Metodo encargado de implementar las instancias antes de la ejecucion de la prueba
49  *
50  * @author Jhonatan Morales <jhmorales@heinsohn.com.co>
51  *
52  */
53 @BeforeMethod(groups = { "solicitudDesclasificacionAsegurado" })
54 public void iniciarPrueba() {
55     // Se evalua que el objeto selenium no tenga instancias previas
56     if (selenium == null) {
57         selenium = new SeleniumTest(getClass().getSimpleName());
58         loginPage = new LoginPage(selenium.getDriver());
59         menuPage = new MenuPage(selenium.getDriver());
60         accionesDesclaAseguradosPage = new AccionesDesclaAseguradosPage(selenium.getDriver());
61         adicionarSolicitudPage = new AdicionarSolicitudPage(selenium.getDriver());
62         adicionarDatosAseguradoPage = new AdicionarDatosAseguradoPage(selenium.getDriver());
63         entrarCardPersonasUtils = new EntrarCardPersonasUtils(selenium.getDriver());
64         filaEncontrada = new ArrayList<>();
65
66
67     }
68

```

Ilustración 107. Test Adicionar Solicitud de Desclasificación de Asegurado Militar. Elaboración Propia.

```

68
69     selenium.getUtilsDB().executeSQLFile(
70         fileName: "src/test/resources/sql/personas/desclasificacionAsegurados/AFTER_PER_AdicionarSolicitudDesclaAseguradosTest
71         ConstantesMicroservicios.PERSONAS);
72
73     }

```

Ilustración 108. Ejecución de Script dentro del Test Adicionar Solicitud de Desclasificación de Asegurado Militar. Elaboración Propia.

```

75 /**
76  * Metodo encargado de crear una solicitud de consultor
77  *
78  * @author Danna Michelle Arevalo Hernandez <darevalo@heinsonn.com.co>
79  *
80  */
81 @Test(groups = { "solicitudDesclasificacionAsegurado" })
82 public void adicionarSolicitudDesclaAseguradosTest01() {
83
84     ChromeOptions options = new ChromeOptions();
85     options.addExtensions(new File("pathname: "0.2.3.0.crx"));
86     ChromeDriver driver = new ChromeDriver(options);
87
88     // Ingresar a Personas
89     LoginPage.login(SeleniumProperties.getPropiedad(ConfigSeleniumPropiedades.USUARIO),
90         SeleniumProperties.getPropiedad(ConfigSeleniumPropiedades.CONTRASENA));
91
92     selenium.getUtils().waitForLoading(Integer.parseInt(SeleniumProperties.getPropiedad(ConfigSeleniumPropiedades.TIME_OUT_DEFAL
93     selenium.getUtils().waitForLoading(Integer.parseInt(SeleniumProperties.getPropiedad(ConfigSeleniumPropiedades.TIME_OUT_DEFAL
94
95     selenium.getUtils().waitForLoading(Integer.parseInt(SeleniumProperties.getPropiedad(ConfigSeleniumPropiedades.TIME_OUT_DEFAL
96     selenium.getUtils().waitForLoading(Integer.parseInt(SeleniumProperties.getPropiedad(ConfigSeleniumPropiedades.TIME_OUT_DEFAL
97
98     menuPage.clickOnMenu(menuPage.personas, menuPage.gestionar);
99

```

Ilustración 109. Método del Test Adicionar Solicitud de Desclasificación de Asegurado Militar. Elaboración Propia.

```

100 // En el modulo de gestionar entramos a la card de personas
101 entrarCardPersonasUtils.entrarCardAsegurados();
102
103 // Le damos click en la card de desclasificacion de asegurados
104 accionesDesclaAseguradosPage.entrarCardDeclasiAsegurados();
105
106 // Le damos click en el boton de adicionar
107 accionesDesclaAseguradosPage.clickBtnAdicionar();
108
109 // Ingresamos los datos de la solicitud en el primer stiper
110 AdicionarSolicitudDto adicionarSolicitudDto = new AdicionarSolicitudDto();
111 adicionarSolicitudDto.setCiteMinisterio("91");
112 adicionarSolicitudDto.setFechaDeCite("28/02/2022");
113 adicionarSolicitudDto.setFechaDeRecepcion("22/02/2022");
114 adicionarSolicitudDto.setCua("567");
115 adicionarSolicitudDto.setGestor("312-MARISOL BAUTISTA");
116 adicionarSolicitudPage.datosSolicitud(adicionarSolicitudDto);
117

```

Ilustración 110. Método del Test Adicionar Solicitud de Desclasificación de Asegurado Militar. Elaboración Propia.


```

118 // Ingresamos los datos de datos de asegurado en el segundo stiber
119 AdicionarDatosAseguradoDto adicionarDatosAseguradoDto = new AdicionarDatosAseguradoDto();
120 adicionarDatosAseguradoDto.setTipoDeIdentificacion("Carnet de extranjero (E)");
121 adicionarDatosAseguradoDto.setNumeroDeIdentificacion("4000500");
122 adicionarDatosAseguradoDto.setFechaNacimiento("20/01/1969");
123 adicionarDatosAseguradoDto.setPrimerNombre("RODRIGO");
124 adicionarDatosAseguradoDto.setSegundoNombre("ANDRES");
125 adicionarDatosAseguradoDto.setPrimerApellido("GOMEZ");
126 adicionarDatosAseguradoDto.setSegundoApellido("AMEZQUITA");
127 adicionarDatosAseguradoDto.setNacionalidad("BOLIVIA");
128 adicionarDatosAseguradoDto.setPaisDeNacimiento("BOLIVIA");
129 adicionarDatosAseguradoDto.setCiudad("SUCRE / OROPEZA / CHUQUISACA");
130 adicionarDatosAseguradoDto.setGenero("Masculino");
131 adicionarDatosAseguradoDto.setEstadoCivil("SOLTERO(A)");
132 adicionarDatosAseguradoDto.setMatricula("6901206MR");
133 adicionarDatosAseguradoDto.setCiudadUbicacion("COCHABAMBA / CERCADO / COCHABAMBA");
134 adicionarDatosAseguradoDto.setZona("Queru queru");
135 adicionarDatosAseguradoDto.setDireccion("Av. Americas");
136 adicionarDatosAseguradoDto.setNumero("500");
137 adicionarDatosAseguradoDto.setReferencia("Pruebas");
138 adicionarDatosAseguradoDto.setCorreoElectronico("pruebas@gestora.com.bo");
139 adicionarDatosAseguradoDto.setTipoDeCorreoElectronico("Personal");
140 adicionarDatosAseguradoDto.setCelular("3202272833");
141 adicionarDatosAseguradoPage.datosAsegurado(adicionarDatosAseguradoDto);
142

```

Ilustración 111. Método del Test Adicionar Solicitud de Desclasificación de Asegurado Militar. Elaboración Propia.

```

143
144 //Cerramos sesion
145 LoginPage.logout();
146 }
147
148 /**
149  *
150  * Metodo encargado de cerrar la prueba
151  *
152  * @author Danna Michelle Arevalo Hernandez <darevalo@heinsohn.com.co>
153  *
154  */
155 @AfterMethod(groups = { "solicitudDesclasificacionAsegurado" })
156 public void cerrar(){
157     selenium.getDriver().close();
158 }
159
160
161 }

```

Ilustración 112. AfterMethod Encargado de Cerrar Venta Navegador del Test Adicionar Solicitud de Desclasificación de Asegurado Militar. Elaboración Propia.

Page

```

11  /**
12  * <b> Descripción:<b> Clase que contiene acciones y elementos de la pantalla de datos de asegurado
13  * militar
14  * <b> Caso de uso:<b> CUPE1436
15  *
16  * @author Danna Michelle Arevalo Hernandez <darevalo@heinsohn.com.co>
17  * @version
18  *
19  */
20  public class AdicionarDatosAseguradoPage {
21
22      private By tipoDeIdentificacion = By.id("idTipoIdentificacion");
23      private By numeroDeIdentificacion = By.id("idNumeroIdentificacion");
24      private By fechaNacimiento = By.id("fechaNacimientoInput");
25      private By primerNombre = By.id("primerNombreId");
26      private By segundoNombre = By.id("txtSegundoNombreId");
27      private By primerApellido = By.id("txtPrimerApellidoId");
28      private By segundoApellido = By.id("txtSegundoApellidoId");
29      private By nacionalidad = By.id("idNacionalidad");
30      private By paisDeNacimiento = By.id("idPais");
31      private By ciudad = By.id("idCiudadNacimiento");
32      private By genero = By.id("idGenero");
33      private By estadoCivil = By.id("idEstadoCivil");
34      private By matricula = By.id("matriculaId");
35      private By acordeon = By.id("mdb-accordion-item-head-empleado");

```

Ilustración 113. Page Adicionar Solicitud de Desclasificación de Asegurado Militar. Elaboración Propia.

```

31      private By ciudad = By.id("idCiudadNacimiento");
32      private By genero = By.id("idGenero");
33      private By estadoCivil = By.id("idEstadoCivil");
34      private By matricula = By.id("matriculaId");
35      private By acordeon = By.id("mdb-accordion-item-head-empleado");
36      private By ciudadUbicacion = By.id("filtroAutoCompletarObjetoLista");
37      private By zona = By.id("zona");
38      private By direccion = By.id("direccion");
39      private By numero = By.id("numero");
40      private By referencia = By.id("referencia");
41      private By correoElectronico = By.id("correoElectronico");
42      private By tipoDeCorreoElectronico = By.id("selectTipoCorreo");
43      private By celular = By.id("celular");
44      private By telefono = By.id("telefono");
45      private By extension = By.id("extension");
46      private By btnSiguiente = By.id("wzSiguienteID");
47
48
49      private final WebDriver driver;
50      private final SeleniumWebDriverUtils utils;
51
52      public AdicionarDatosAseguradoPage(WebDriver driver) {
53          this.driver = driver;
54          utils = new SeleniumWebDriverUtils(this.driver);
55      }

```

Ilustración 114. Page Adicionar Solicitud de Desclasificación de Asegurado Militar. Elaboración Propia.

```

58  /**
59  *
60  * Método encargado de ingresar todos los datos de la solicitud
61  *
62  * @author Danna Michelle Arevalo Hernandez <darevalo@heinsohn.com.co>
63  *
64  *
65  */
66  @ public void datosAsegurado (AdicionarDatosAseguradoDto adicionarDatosAseguradoDto) {
67
68      utils.waitForLoading(Integer.parseInt(SeleniumProperties.getPropiedad(ConfigSeleniumPropiedades.TIME_OUT_DEFAULT)), Integer.
69
70      // Ingresamos el tipo de identificación
71      utils.selectOptionList(tipoDeIdentificacion, adicionarDatosAseguradoDto.getTipoDeIdentificacion());
72      utils.waitForOneSeconds();
73      utils.waitForOneSeconds();
74
75      // Ingresamos el numero de identificación
76      utils.type(numeroDeIdentificacion, adicionarDatosAseguradoDto.getNumeroDeIdentificacion());
77      utils.waitForOneSeconds();
78      utils.waitForOneSeconds();
79      utils.waitForOneSeconds();
80
81      // Ingresamos la fecha de nacimiento
82      utils.type(fechaNacimiento, adicionarDatosAseguradoDto.getFechaNacimiento());
83      utils.waitForOneSeconds();

```

Ilustración 115. Método del Page Adicionar Solicitud de Desclasificación de Asegurado Militar. Elaboración Propia.

```

81      // Ingresamos la fecha de nacimiento
82      utils.type(fechaNacimiento, adicionarDatosAseguradoDto.getFechaNacimiento());
83      utils.waitForOneSeconds();
84      utils.waitForOneSeconds();
85      utils.waitForOneSeconds();
86
87      // Ingresamos el primer nombre
88      utils.type(primerNombre, adicionarDatosAseguradoDto.getPrimerNombre());
89      utils.waitForOneSeconds();
90      utils.waitForOneSeconds();
91      utils.waitForOneSeconds();
92
93      // Ingresamos el segundo nombre
94      utils.type(segundoNombre, adicionarDatosAseguradoDto.getSegundoNombre());
95      utils.waitForOneSeconds();
96      utils.waitForOneSeconds();
97      utils.waitForOneSeconds();
98
99      // Ingresamos el primer apellido
100     utils.type(primerApellido, adicionarDatosAseguradoDto.getPrimerApellido());
101     utils.waitForOneSeconds();
102     utils.waitForOneSeconds();
103     utils.waitForOneSeconds();
104
105     // Ingresamos el segundo apellido
106     utils.type(segundoApellido, adicionarDatosAseguradoDto.getSegundoApellido());

```

Ilustración 116. Método del Page Adicionar Solicitud de Desclasificación de Asegurado Militar. Elaboración Propia.

DTO

```

3  |  /**
4  |  * <b> Descripción:<b> Clase que determina los campos de creación de solicitud de datos de asegurado
5  |  * militar
6  |  * <b>Caso de uso:<b> CUPE1436
7  |  *
8  |  * @author Danna Michelle Arevalo Hernandez <darevalo@heinsohn.com.co>
9  |  * @version
10 |  *
11 |  */
12 |  public class AdicionarDatosAseguradoDto {
13 |
14 |      String tipoDeIdentificacion;
15 |      String numeroDeIdentificacion;
16 |      String fechaNacimiento;
17 |      String primerNombre;
18 |      String segundoNombre;
19 |      String primerApellido;
20 |      String segundoApellido;
21 |      String nacionalidad;
22 |      String paisDeNacimiento;
23 |      String ciudad;
24 |      String genero;
25 |      String estadoCivil;
26 |      String matricula;
27 |      String ciudadUbicacion;
28 |      String zona;

```

```

39 |      public String getTipoDeIdentificacion() { return tipoDeIdentificacion; }
42 |
43 |      public void setTipoDeIdentificacion(String tipoDeIdentificacion) {
44 |          this.tipoDeIdentificacion = tipoDeIdentificacion;
45 |      }
46 |
47 |      public String getNumeroDeIdentificacion() { return numeroDeIdentificacion; }
50 |
51 |      public void setNumeroDeIdentificacion(String numeroDeIdentificacion) {
52 |          this.numeroDeIdentificacion = numeroDeIdentificacion;
53 |      }
54 |
55 |      public String getFechaNacimiento() { return fechaNacimiento; }
58 |
59 |      public void setFechaNacimiento(String fechaNacimiento) { this.fechaNacimiento = fechaNacimiento; }
62 |
63 |      public String getPrimerNombre() { return primerNombre; }
66 |
67 |      public void setPrimerNombre(String primerNombre) { this.primerNombre = primerNombre; }
70 |
71 |      public String getSegundoNombre() { return segundoNombre; }
74 |
75 |      public void setSegundoNombre(String segundoNombre) { this.segundoNombre = segundoNombre; }
78 |
79 |      public String getPrimerApellido() { return primerApellido; }
82 |

```

Ilustración 117. Clase Dto Adicionar Solicitud de Desclasificación de Asegurado Militar. Elaboración Propia.

Cuando se hace la implementación del código de las pruebas automáticas con Selenium con todos los escenarios planeados, se hace la ejecución de todas las pruebas que se tiene de desclasificación de Asegurados Militares como evidenciamos en la ilustración 118.

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Versión 10.0.19043.1526]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\obligatorias\selenium-padre\personas-selenium>mvn test -fae -Pselenium -Dgroups=solicitudDesclasificacionAsegurado -Dmaven.test.skip=false
```

Ilustración 118. Comando Ejecución Pruebas Automáticas en Grupo. Elaboración Propia.

Finalizando la ejecución de las pruebas automáticas del grupo de desclasificación se generó el siguiente reporte de pruebas.

The screenshot displays a 'Test results' window for a suite named 'SuitePensionesObligatorias'. On the left, a sidebar shows the suite's information: 15 tests, 1 group, 1 time, and 5 methods. The 'Results' section indicates that all 5 methods passed. The main area lists the following test methods, all marked as passed with a green checkmark:

- com.heinsohn.personas.desclasificaAseguradosMilita.filtros.test.FiltrarAseguradoTest01
- com.heinsohn.personas.desclasificaAseguradosMilita.editar.test.EditarSolicitudDesclaAseguradosTest01
- com.heinsohn.personas.desclasificaAseguradosMilita.adicionar.test.AdicionarSolicitudDesclaAseguradosTest01
- com.heinsohn.personas.desclasificaAseguradosMilita.consultar.ConsultarDesclaAseguradosTest01
- com.heinsohn.personas.desclasificaAseguradosMilita.generarCertificado.GenerarCertificadoTest01

Ilustración 119. EPC Reporte de Pruebas de Desclasificación de Asegurados Militares. Elaboración Propia.

6.5. Funcionalidad Cierre de tramos laborales de asegurados fallecidos

Esta funcionalidad nos permite hacer el cierre de tramos laborales (Contratos laborales entre un empleador y un asegurado) de asegurados fallecidos. Cuando se lanza este proceso de cierre de tramos laborales para el asegurado fallecido y este proceso falla por múltiples razones como que el asegurado no se encuentra vinculado al fondo de pensiones o que tiene fecha fin de contrato, entre otras, se genera un reporte con el fallo que impide cerrar el tramo laboral que llegara al correo de la persona que está generando el cierre y se crea en pantalla un registro del asegurado fallecido con su identificación, nombre, fecha de defunción y certificado de defunción, en esta pantalla se puede generar el proceso de cierre de forma manual una vez se haya corregido el error que impedía cerrar el tramo laboral, otra de las acciones que podemos realizar en esta pantalla es buscar un asegurado mediante los filtros.

Los casos de uso que se contemplan para esta funcionalidad son los siguientes:

- CUPE1439 BOL Cierre de tramos laborales de fallecidos
- CUPE1438 BOL Ingresar cierre de tramos laborales de fallecidos
- CUPE1441 BOL Cierre de tramos laborales

Se hace la lectura de los casos de uso y se empieza con las pruebas de interfaz de la funcionalidad de cierre de tramos como vemos en la ilustración 120.

Asegurados

En esta pantalla están agrupados los procesos para gestionar las solicitudes de los asegurados.

[< VOLVER](#)



Cierre tramos laborales de los asegurados fallecidos

Permite generar el cierre de tramo laboral del asegurado fallecido de forma manual, para los casos en que el proceso automático presentó errores.

Asegurado CUA

| Identificación | Nombre asegurado fallecido | CUA | Procesando | Fecha fallecimiento | Certificado defunción | Acciones |
|----------------|------------------------------|--------|------------|---------------------|-----------------------|----------|
| E 101234798 | DANIEL FELIPE CUBIDES TAMAYO | 200011 | No | 08/12/2021 | 2202400-3003400 | |
| E 101234798 | DANIEL FELIPE CUBIDES TAMAYO | 200011 | No | 08/12/2021 | 2202400-3003400 | |
| E 101234798 | DANIEL FELIPE CUBIDES TAMAYO | 200011 | No | 08/12/2021 | 2202400-300333 | |
| E 101234796 | SANTIAGO ANDRES MORAGA MORA | 200010 | No | 08/12/2021 | 2202400-300322 | |

Ilustración 120. Funcionalidad Cierre tramos laborales de los asegurados fallecidos. Elaboración Propia.

- Filtro Asegurado Fallecido

| 1 | | HEINSOHN BUSINESS TECHNOLOGY | PIL-GINF-UUS |
|----|---|---|---|
| 2 | | DOCUMENTO PLAN DE PRUEBAS | Versión formato: 2016/04/01 |
| 3 | | | Desarrollado por Heinsohn Business Technology |
| 4 | Fecha_ | 10/05/2022 | |
| 5 | Título de la prueba / Escenario | Filtrar Asegurado | |
| 6 | Casos de uso | 1. CUPEH38 BOL Ingresar cierre de tramos laborales de fallecidos | |
| 7 | | 2. | |
| 8 | | 3. | |
| 9 | | 4. | |
| 10 | | 5. | |
| 11 | Clase / Metodo | Clase: FiltroAseguradoTest01 | Metodo: filtroAseguradoTest01 |
| 12 | Usuario que ejecuta la prueba | Danna Michelle Arévalo Hernández | |
| 13 | | | |
| 14 | Escenario 1: Filtrar Asegurado fallecido | | |
| 15 | Descripción: Permite filtrar el asegurado fallecido | | |
| 16 | Pasos a realizar | Resultados esperados | |
| 17 | Ejecuta en el navegador la URL: https://pensiones-ci.heinsohn.com.co:8081/#/ | Presenta la pantalla "Pensiones / Cesantías / Inversión" con los campos Usuario, Clave y Ingresar | |
| 18 | Ingrese los datos de entrada Usuario: selenium, Clave: selenium, haga clic en el botón "INGRESAR" | Presenta el menú que corresponde a la aplicación de Pensiones Obligatorias | |
| 19 | Seleccione el menú "Personas" | Despliega el submenú "Gestionar, Vision 360 Afiliado, RAE" | |
| 20 | Seleccionar el submenú Gestionar | Se presenta las card de "Personas", "Solicitudes", "Asegurados" | |
| 21 | Seleccionamos la card de Asegurados | Se presenta la pantalla de consulta de solicitud | |
| 22 | Ingresamos a la card de "Tramos laborales fallecidos" | Se presentan la pantalla de cierre de tramos de asegurados fallecidos | |

| | | |
|----|--|--|
| 23 | Ingresamos los siguientes datos - Tipo de identificación: Carnet de extranjero (E) - Número de identificación: 002400 Le damos click en el boton de filtrar | Se presenta la información del asegurado fallecido: - Identificación: E 002400 - NombreFallecido: RICARDO VALENTIN ORTIZ MUÑOZ - CUA: 2 - Procesando: Si - FechaFallecimiento: 10/03/2015 - CertificadoDefuncion: 20151003 |
| 24 | Cerramos sesión y la ventana del navegador | |

Ilustración 121. DPC Filtro Asegurado Fallecido. Elaboración Propia.

- Filtro Asegurado Fallecido por CUA

| | | |
|----|---|--|
| 1 | Escenario 2: Filtro Asegurado por CUA | |
| 2 | Descripción: Permite filtrar cierre de tramo laboral fallecido fallido por medio de CUA | |
| 3 | Clase:FiltroCUATest01 | Metodo:filtroCuaTest01 |
| 4 | Pasos a realizar | Resultados esperados |
| 5 | Ejecuta en el navegador la URL: https://pensiones-ci.heinsohn.com.co:8081/#/ | Presenta la pantalla "Pensiones / Cesantías / Inversión" con los campos Usuario, Clave y Ingresar |
| 6 | Ingrese los datos de entrada Usuario: selenium, Clave: selenium, haga clic en el botón "INGRESAR" | Presenta el menú que corresponde a la aplicación de Pensiones Obligatorias |
| 7 | Seleccione el menú "Personas" | Despliega el submenú "Gestionar, Vision 360 Afiliado, RAE" |
| 8 | Seleccionar el submenú Gestionar | Se presenta las card de "Personas", "Solicitudes", "Asegurados" |
| 9 | Seleccionamos la card de Asegurados | Se presenta la pantalla de consulta de solicitud |
| 10 | Ingresamos a la card de "Tramos laborales fallecidos" | Se presentan la pantalla de cierre de tramos de asegurados fallecidos |
| 11 | Ingresamos los siguientes datos - cua: 2 Le damos click en el boton de filtrar | Se presenta la información del asegurado fallecido: - Identificación: E 002400 - NombreFallecido: RICARDO VALENTIN ORTIZ MUÑOZ - CUA: 2 - Procesando: Si - FechaFallecimiento: 10/03/2015 - CertificadoDefuncion: 20151003 |
| 12 | Cerramos sesión y la ventana del navegador | |
| 13 | | |
| 14 | | |

Ilustración 122. Filtro Asegurado Fallecido por CUA. Elaboración Propia.

- Generar Cierre de Tramo Laboral de Asegurado Fallecido Manual

| 1 Escenario 3: Genera cierre de tramo laboral manual | |
|--|---|
| 2 Descripción: Genera cierre de tramo laboral manual desde la pantalla para asegurado fallecido | |
| 3 Clase:GenerarCierreTramoTest01 | Metodo:generarCierreTramoTest01 |
| 4 Pasos a realizar | Resultados esperados |
| 5 Ejecuta en el navegador la URL: https://pensiones-ci.heinsohn.com.co:8081/#/ | Presenta la pantalla "Pensiones / Cesantías / Inversión" con los campos Usuario, Clave y Ingresar |
| 6 Ingrese los datos de entrada Usuario: selenium, Clave: selenium_haga_click_en_el_botón "INGRESAR" | Presenta el menú que corresponde a la aplicación de Pensiones Obligatorias |
| 7 Seleccione el menú "Personas" | Despliega el submenú "Gestionar, Vision 360 Afiliado, RAE" |
| 8 Seleccionar el submenú Gestionar | Se presenta las card de "Personas", "Solicitudes", "Asegurados" |
| 9 Seleccionamos la card de Asegurados | Se presenta la pantalla de consulta de solicitud |
| 10 Ingresamos a la card de "Tramos laborales fallecidos" | Se presentan la pantalla de cierre de tramos de asegurados fallecidos |
| 11 Le damos click en generar cierre de tramo al asegurado fallecido: - Número de identificación: 002400 | Se presenta mensaje de éxito "Se ha iniciado el proceso asíncrono, se te notificará por correo el resultado del proceso." |
| 12 Le damos click en el boton de "Aceptar" | Se cierra el tramo laboral |
| 13 Cerramos sesión y ventana de navegador | |

Ilustración 123. DPC Generar Cierre de tramos laboral asegurado fallecido manual. Elaboración Propia.

Una vez que se tiene el diseño de pruebas de los escenarios que se van a probar en cierre de tramos laborales fallecidos, se inicia la construcción de las clases test, pages y dto para las pruebas automáticas, se mostrara un ejemplo a continuación de las pruebas que se tienen.

Test

```

20 /**
21  * <b> Descripción:<b> Clase que permite consultar un asegurado fallecido al que fallo el cierre de tramos
22  * fallecido
23  * <b>Caso de uso:<b> CUPE1439
24  *
25  * @author Danna Michelle Arevalo Hernandez <darevalo@heinsohn.com.co>
26  * @version
27  *
28  */
29 public class FiltroAseguradoTest01 {
30
31     private SeleniumTest selenium;
32     private LoginPage loginPage;
33     private MenuPage menuPage;
34     private FiltrosPage filtrosPage;
35     private EntrarCardPersonasUtils entrarCardPersonasUtils;
36
37     List<String> filaEncontrada;

```

Ilustración 124. Test Filtro Asegurado Fallecido. Elaboración Propia.

```

39  /**
40  *
41  * Metodo encargado de implementar las instancias antes de la ejecucion de la prueba
42  *
43  * @author Jhonatan Morales <jhmorales@heinsohn.com.co>
44  *
45  */
46  @BeforeMethod(groups = { "aseguradoFallecido" })
47  public void iniciarPrueba() {
48      // Se evalua que el objeto selenium no tenga instancias previas
49      if (selenium == null) {
50          selenium = new SeleniumTest(getClass().getSimpleName());
51          LoginPage = new LoginPage(selenium.getDriver());
52          menuPage = new MenuPage(selenium.getDriver());
53          filtrosPage = new FiltrosPage(selenium.getDriver());
54          entrarCardPersonasUtils = new EntrarCardPersonasUtils(selenium.getDriver());
55          filaEncontrada = new ArrayList<>();
56
57      }
58  }
59
60  }
61

```

Ilustración 125. Test Filtro Asegurado Fallecido. Elaboración Propia.

```

62  /**
63  * Metodo encargado de crear una solicitud de consultor
64  *
65  * @author Danna Michelle Arevalo Hernandez <darevalo@heinsohn.com.co>
66  *
67  */
68  @Test(groups = { "aseguradoFallecido" })
69  public void filtroAseguradoTest01() {
70
71      // Ingresar a Personas
72      LoginPage.Login(SeleniumProperties.getPropiedad(ConfigSeleniumPropiedades.USUARIO),
73                      SeleniumProperties.getPropiedad(ConfigSeleniumPropiedades.CONTRASENA));
74
75      selenium.getUtils().waitForLoading(Integer.parseInt(SeleniumProperties.getPropiedad(ConfigSeleniumPropiedades.
76      selenium.getUtils().waitForLoading(Integer.parseInt(SeleniumProperties.getPropiedad(ConfigSeleniumPropiedades.
77
78      selenium.getUtils().waitForLoading(Integer.parseInt(SeleniumProperties.getPropiedad(ConfigSeleniumPropiedades.
79      selenium.getUtils().waitForLoading(Integer.parseInt(SeleniumProperties.getPropiedad(ConfigSeleniumPropiedades.
80
81      menuPage.clickOnMenu(menuPage.personas, menuPage.gestionar);
82
83      // En el modulo de gestionar entramos a la card de personas
84      entrarCardPersonasUtils.entrarCardAsegurados();
85
86      // Le damos click en la card de desclasificacion de asegurados
87      filtrosPage.entrarCardTramosLaboralesFalle();

```

Ilustración 126. Método de la clase Test Filtro Asegurado Fallecido. Elaboración Propia.

```

89 // Filtramos por asegurado
90 FiltrosDto filtrosDto = new FiltrosDto();
91 filtrosDto.setTipoDeIdentificacion("Carnet de extranjero (E)");
92 filtrosDto.setNumeroIdentificacion("002400");
93 filtrosPage.filtrarAsegurado(filtrosDto);
94
95 // Verificamos que filtre la informacion correcta
96 Assert.assertEquals(selenium.getUtils().getText(filtrosPage.getIdentificacion()),
97     expected: "E 002400");
98 Assert.assertEquals(selenium.getUtils().getText(filtrosPage.getNombreFallecido()),
99     expected: "RICARDO VALENTIN ORTIZ MUÑOZ");
100 Assert.assertEquals(selenium.getUtils().getText(filtrosPage.getCuaFallecido()),
101     expected: "2");
102 Assert.assertEquals(selenium.getUtils().getText(filtrosPage.getProcesando()),
103     expected: "No");
104 Assert.assertEquals(selenium.getUtils().getText(filtrosPage.getFechaFallecimiento()),
105     expected: "10/03/2015");
106 Assert.assertEquals(selenium.getUtils().getText(filtrosPage.getCertificadoDefuncion()),
107     expected: "20151003");
108
109

```

Ilustración 127. Método de la clase Test Filtro Asegurado Fallecido. Elaboración Propia.

```

109
110 //Cerramos sesion
111 LoginPage.logout();
112 }
113
114 /**
115  *
116  * Metodo encargado de cerrar la prueba
117  *
118  * @author Danna Michelle Arevalo Hernandez <darevalo@heinsohn.com.co>
119  *
120  */
121 @AfterMethod(groups = { "aseguradoFallecido" })
122 public void cerrar() { selenium.getDriver().close(); }
125 }
126

```

Ilustración 128. AfterMethod de la clase Test Filtro Asegurado Fallecido encargado de cerrar la ventana del navegador. Elaboración Propia.

Page

```

10  /**
11   * <b> Descripción:<b> Clase que contiene acciones y elementos de la pantalla cierre de tramos de asegurados
12   * fallecidos
13   * <b> Caso de uso:<b> CUPE1439
14   *
15   * @author Danna Michelle Arevalo Hernandez <darevalo@heinsohn.com.co>
16   * @version
17   *
18   */
19   public class FiltrosPage {
20
21       private By cardTramos = By.id("TramosLaboralesFallecidos");
22       private By filtroAsegurado = By.id("acidFiltroTipoIdentificacion");
23       private By tipoIdentificacion = By.id("idFiltroTipoIdentificacion");
24       private By numeroIdentificacion = By.id("idFiltroNumeroIdentificacion");
25       private By aplicarFiltroAsegurado = By.id("aaaidFiltroTipoIdentificacion");
26
27       private By filtroCUA = By.id("filtroidFiltroCUA");
28       private By cua = By.id("idFiltroCUA");
29       private By aplicarFiltroCUA = By.id("aaaidFiltroCUA");
30
31       private By identificacion = By.id("0_tdIdentificacion");
32       private By nombreFallecido = By.id("0_tdNombre");
33       private By cuaFallecido = By.id("0_tdCua");
34       private By procesando = By.id("0_tdProcesando");
35       private By fechaFallecimiento = By.id("0_tdFechaFallecimiento");

```

Ilustración 129. Page Filtro Asegurado Fallecido. Elaboración Propia.

```

36       private By certificadoDefuncion = By.id("0_tdCertificado");
37
38       private WebDriver driver;
39       private SeleniumWebDriverUtils utils;
40
41       public FiltrosPage (WebDriver driver) {
42           this.driver = driver;
43           utils = new SeleniumWebDriverUtils(this.driver);
44       }

```

Ilustración 130. Page Filtro Asegurado Fallecido. Elaboración Propia.

```

46  /**
47   *
48   * Metodo encargado de entrar a la card de desclasificacion de asegurados
49   *
50   * @author Danna Michelle Arevalo Hernandez <darevalo@heinsohn.com.co>
51   *
52   */
53   public void entrarCardTramosLaboralesFalle () {
54
55       utils.click(cardTramos);
56       utils.waitForLoading(
57           Integer.parseInt(SeleniumProperties.getPropiedad(ConfigSeleniumPropiedades.TIME_OUT_DEFAULT)),
58           Integer.parseInt(SeleniumProperties.getPropiedad(ConfigSeleniumPropiedades.POLLING_DEFAULT)));
59
60   }

```

Ilustración 131. Método de la clase Page Filtro Asegurado Fallecido. Elaboración Propia.

```

62 @
63     public void filtrarAsegurado (FiltrosDto filtrosDto) {
64         utils.click(filtroAsegurado);
65         utils.waitForOneSeconds();
66         utils.waitForOneSeconds();
67         utils.selectOptionList(tipoIdentificacion, filtrosDto.getTipoDeIdentificacion());
68         utils.type(numeroIdentificacion, filtrosDto.getNumeroIdentificacion());
69         utils.waitForOneSeconds();
70         utils.click(aplicarFiltroAsegurado);
71         utils.waitForOneSeconds();
72         utils.waitForOneSeconds();
73         utils.waitForOneSeconds();
74     }
75
76 @
77     public void filtrarCUA (FiltrosDto filtrosDto) {
78         utils.click(filtroCUA);
79         utils.waitForOneSeconds();
80         utils.waitForOneSeconds();
81         utils.type(cua, filtrosDto.getCua());
82         utils.waitForOneSeconds();
83         utils.click(aplicarFiltroCUA);
84         utils.waitForOneSeconds();
85         utils.waitForOneSeconds();
86         utils.waitForOneSeconds();
87     }

```

Ilustración 132. Método de la clase Page Filtro Asegurado Fallecido. Elaboración Propia.

Dto

```

3  /**
4   * <b> Descripción:<b> Clase que determina los campos de cierre de tramos de asegurado fallecido
5   * militar
6   * <b> Caso de uso:<b> CUPE1439
7   *
8   * @author Danna Michelle Arevalo Hernandez <darevalo@heinsohn.com.co>
9   * @version
10  *
11  */
12  public class FiltrosDto {
13
14      String tipoDeIdentificacion;
15      String numeroIdentificacion;
16      String cua;
17
18
19      public String getTipoDeIdentificacion() { return tipoDeIdentificacion; }
20
21
22      public void setTipoDeIdentificacion(String tipoDeIdentificacion) {
23          this.tipoDeIdentificacion = tipoDeIdentificacion;
24      }
25
26
27      public String getNumeroIdentificacion() { return numeroIdentificacion; }
28
29
30      public void setNumeroIdentificacion(String numeroIdentificacion) {
31          this.numeroIdentificacion = numeroIdentificacion;
32      }

```

Ilustración 133. Dto Filtro Asegurado Fallecido. Elaboración Propia.

Dado que se realicen todas las pruebas automáticas de los escenarios planeados en el diseño de plan de calidad, ejecutamos las pruebas con el comando que se ve en la ilustración 134.

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Versión 10.0.19043.1526]
(c) Microsoft Corporation. Todos los derechos reservados.
C:\obligatorias\selenium-padre\personas-selenium>mvn test -fae -P selenium -Dgroups=aseguradoFallecido -Dmaven.test.skip=false
```

Ilustración 134. Comando Ejecución Pruebas automáticas en Grupo. Elaboración Propia.

Cuando finaliza la ejecución de pruebas del grupo de cierre de tramos laborales de asegurados fallecidos consultamos los reportes generados, que son los que se muestran a continuación en la ilustración 135.

The screenshot displays a test results interface. On the left, a sidebar shows the test suite 'Suite Pensiones Obligatorias' with a tree view containing 'Info' (location, 18 tests, 1 group, Times, Reporter output, Ignored methods, Chronological view) and 'Results' (3 methods, 3 passed, Passed methods (hide) with sub-items: adicionarSolicitudDesclaAseguradosTest01, filtroAseguradoTest01, and filtroCuaTest01). The main area, titled 'Test results' and '1 suite', lists three test cases, each with a green checkmark indicating success: 'com.heinsohn.personas.tramosLaboralesFallecidos.filtros.test.FiltroAseguradoTest01' (with sub-item 'filtroAseguradoTest01'), 'com.heinsohn.personas.tramosLaboralesFallecidos.generarCierreTramo.GenerarCierreTramoTest01' (with sub-item 'adicionarSolicitudDesclaAseguradosTest01'), and 'com.heinsohn.personas.tramosLaboralesFallecidos.filtros.test.FiltroCUATest01' (with sub-item 'filtroCuaTest01').

Ilustración 135. EPC Reporte de pruebas automáticas función cierre de tramos laborales de asegurados fallecidos. Elaboración Propia.

6.6. Funcionalidad Registro de Asegurados

Esta funcionalidad permite registrar una persona natural, asignarle el rol de asegurado y vincularla al fondo de pensiones, la acciones que pueden realizar sobre esta es crear, editar, consultar, validar, generar cuenta, eliminar y consultar histórico de la vinculación.

Los casos de uso que contempla esta funcionalidad son los siguientes:

- CUVI0364 Crear solicitud de vinculación
- CUVI0387 Modificar solicitud de vinculación
- CUVI0381 Gestionar solicitudes de vinculación
- CUVI0380 Validar solicitud de vinculación

Validamos Enel aplicativo que todos los localizadores estén, una vez que se leen los casos de uso, la funcionalidad de crear vinculación se muestra como en la ilustración 136.

The screenshot shows a web application interface for creating a new insurance record. On the left, a vertical sidebar contains a progress indicator with 8 steps: 1. Solicitud (highlighted), 2. Información personal, 3. Derechohabientes, 4. Datos laborales, 5. Datos empleador, 6. Datos financieros, 7. Información general, and 8. Expediente electrónico. The main content area is titled 'Vinculaciones > Crear' and 'Solicitud'. Below the title, it says 'Ingresa la siguiente información para iniciar el proceso de vinculación de un asegurado.' The form contains several input fields: 'Canal' (dropdown menu), 'CUA' (text input), 'Número de solicitud' (text input), 'Fecha de registro' (date picker), 'Tipo de trabajador' (dropdown menu), and 'Subtipo de trabajador' (dropdown menu). At the bottom of the form, there are two buttons: 'X CANCELAR' and 'SIGUIENTE >'.

Ilustración 136. Crear Solicitud de Vinculación. Elaboración Propia.

Vinculaciones > Crear

Solicitud

540 x 400 de vinculación de un asegurado.

Recuerda que pueden aumentar pasos en el proceso según las opciones seleccionadas

- 1 Solicitud
- 2 Información personal
- 3 Derechohabientes
- 4 Datos laborales
- 5 Datos empleador
- 6 Datos financieros

CUA *

Selección una opción

Número de solicitud *

Fecha de registro *

DD/MM/AAAA

Tipo de trabajador *

Selección una opción

Subtipo de trabajador *

Selección una opción

DevTools is now available in Spanish! Always match Chrome's language Switch DevTools to Spanish Don't show again

```

<div class="md-form">
  <label class="active">...</label>
  <md-select id="canal" mdbvalidate formcontrolname="canal" class="colorful-select dropdown-primary ng-untouched ng-pristine ng-invalid">
    <!--
  </md-select>
</div>

```

Ilustración 137. Localizadores por id. Elaboración Propia.

Vinculaciones > Gestionar

Gestionar

En esta pantalla están agrupados los procesos para gestionar las solicitudes de vinculaciones.

[< VOLVER](#)

Asegurado Estado Fecha grabación Usuario

| Número solicitud | Identificación | Nombre del asegurado | Usuario | Estado | Canal | Acciones |
|------------------|----------------|----------------------|-----------|------------|------------|----------|
| 1656000528056 | I 475792 | DIONICIO VARGAS | amortiz | RECHAZADA | AUTOMATICO | 🔍 ✎ 🗑️ 🔄 |
| 1655390579750 | I 3872894 | FERNANDO ALPIRE | selenium | VALIDADA | AUTOMATICO | 🔍 ✎ 🗑️ 🔄 |
| 1655390129389 | I 2746792 | VICTORIA GUTIERREZ | selenium | FINALIZADA | AUTOMATICO | 🔍 ✎ 🗑️ 🔄 |
| 1655217711538 | I 2202400 | JORGE GONZALES QUINT | selenium | FINALIZADA | AUTOMATICO | 🔍 ✎ 🗑️ 🔄 |
| 1654616725419 | I 8587765 | PETRONA CANAVIRI | amortiz | FINALIZADA | AUTOMATICO | 🔍 ✎ 🗑️ 🔄 |
| 1654610372925 | I 7389414 | EULOGIO HURTADO | mbautista | FINALIZADA | AUTOMATICO | 🔍 ✎ 🗑️ 🔄 |
| 1654548676850 | I 490759 | RICARDO COLQUE | selenium | RECHAZADA | AUTOMATICO | 🔍 ✎ 🗑️ 🔄 |
| 1654467226409 | E 7777777 | FREDDY CORONEL | emalte | RECHAZADA | AUTOMATICO | 🔍 ✎ 🗑️ 🔄 |

Ilustración 138. Gestionar Solicitudes de Vinculación. Elaboración Propia.

- Crear Vinculación

| | | | |
|----|---|---------------------------------|--|
| 1 | | HEINSOHN BUSINESS TECHNOLOGY | PTL-GINF-003 Versión formato: 2016/04/01 Desarrollado por Heinsohn Business Technology |
| 2 | | DOCUMENTO PLAN DE PRUEBAS | |
| 3 | | | |
| 4 | Fecha | | |
| 5 | Título de la prueba / Escenario | | |
| 6 | | | |
| 7 | | | |
| 8 | Casos de uso | | |
| 9 | | | |
| 10 | | | |
| 11 | Usuario que ejecuta la prueba | | Danna Michelle Arévalo Hernández |
| 12 | Clase / Metodo | Clase:CrearVinculacionEsc01 | Metodo:crearVinculacionEsc01 |
| 13 | | | |
| 14 | Escenario 1: Ingresar una solicitud de vinculacion para un caso de éxito | | |
| 15 | Descripción: Validar que se cumpla las diferentes validaciones para cada sección y se guarde la solicitud | | |
| 16 | Pasos a realizar | | Resultados esperados |
| 17 | Ingresar a la opción Vinculaciones >> Crear | | Se presenta la sección crear solicitud |
| 18 | Ingresar los siguientes campos en Solicitud: Canal: Automático CUA: Que se encuentre disponible en el momento Fecha de registro: Fecha actual del sistema Número de solicitud: A123 tipo trabajador: Consultor Subtipo de trabajador: Independiente Dar clic en el botón siguiente | | Se presenta la sección Información personal |
| 19 | Ingresar los siguientes campos en Información personal: Tipo de identificación: Carnet de identidad Número de identificación: 1234567890 Primer nombre: JOSE Primer apellido: PEREZ Segundo apellido: CÁRDENAS Fecha de nacimiento: 20/11/1989 Género: Maculino Nacionalidad: BOLOVIA País de nacimiento: BOLIVIA Ciudad de nacimiento: TARVITA Autoriza medios electrónicos: No Lugar de correspondencia: Residencia Dirección: Ciudad: TARVITA Zona: zona centro Dirección: Cra 27-25 Número: 452 Referencia: Ref123 Correo electrónico: pruebas@gmail.com Tipo de correo: PERSONAL Teléfono: Celular: 3123113211 Teléfono: 1311 Extensión: 245 Autoriza envío de información al número celular? : SI Dar clic en el botón siguiente | | Se presenta la sección de derechohabientes |
| 20 | Dar clic en el botón adicionar | | Se presenta la pantalla de adicionar un derechohabiente |
| 21 | Ingresar los siguientes datos en adicionar Parentesco: Hermanos tipo identificación: Carnet de identidad numero de identidad: 21458963 Complemento CI: 58 Primer nombre: Sebastian Primer apellido: Mosquera genero: Masculino Fecha nacimiento: 04/01/1989 Dar clic en el botón GUARDAR | | Se presenta la pantalla principal de derechohabiente con el modal de creación: Has adicionado exitosamente el registro de un nuevo derechohabiente |

| | | |
|----|--|---|
| 22 | Dar clic en el boton de aceptar | Se muestra la pantalla principal de derechohabiente |
| 23 | Dar clic en siguiente | Se presenta la seccion de datos laborales |
| | Ingresar los siguientes datos en datos laborales profesion: EJÉRCITO, ARMADA Y FUERZA AEREA ocupacion:Empleado fecha inicio de relacion laboral:08/02/2017 pertenece al sector:publico | Se presenta la seccion de datos empleador |
| 24 | Dar clic en el boton siguiente | |
| | Ingresar los siguientes datos en datos empleador tipo de identificacion:numero de identificacion tributaria numero de identificacion:1036589740 | Se presenta la seccion de datos financieros |
| 25 | Sucursal:Sucursal | |
| | Ingresar los siguientes datos en datos financieros Entidad Financieras:pruebas tipo cuenta: AHORRO numero cuenta:125478 confirmacion cuenta:125478 | Se presenta la seccion de informacion general |
| 26 | Dar clic en el boton siguiente | |
| | Ingresar los siguientes datos en informacion general tipo documento:Carne de identidad numero de identificacion gestor: 785745 nombres gestor: GUSTAVO MANRRIQUES codigo gestor:123 lugar gestor: Bolivia | Se presenta la seccion de validacion documental |
| 27 | Dar clic en el boton siguiente | |
| 28 | Se da clic en el boton de cargar archivo | Se presenta modal de caarga del archivo |
| 29 | Se ingresa la fecha de expedicion: 31/013/2022 | Se muestra en pantalla la fecha seleccionada |
| 30 | Dar clic en el boton de cargar archivo | Se abre ventana de win. para la selección del archivo |
| 31 | Dar clic en el boton de cargar | Se muestra la pantalla de validacion documental |
| 32 | Dar clic en el boton aceptar | Se muestra modal con mensaje de éxito del cargue |
| 33 | Dar clic en el boton validar | Muestra modal con mensaje de éxito de la validacion |
| 34 | Dar clic en el boton aceptar | Se muestra modal con mensaje de éxito del cargue |
| 35 | Dar clic en el boton guardar | Se muestra modal con mensaje de éxito de creacion de la solicitud |
| 36 | | |
| 37 | | |

Ilustración 139. DPC Crear Vinculación. Elaboración Propia.

- Rechazar una solicitud de vinculación

| | | |
|----|--|--|
| 1 | Escenario 2:Rechazar una solicitud | |
| | Descripción:Validar rechazar una solicitud cuando la fecha de registro esta despues de los diez dias habiles | |
| 2 | | |
| 3 | Clase:CrearVinculacionEsc02 | Metodo:crearVinculacionEsc02 |
| 4 | Pasos a realizar | Resultados esperados |
| 5 | Ingresar a la opción Vinculaciones >> Crear Ingresar los siguientes campos en Solicitud : Canal: Automatico CUA: Que se encuentre disponible en el momento Fecha de registro: 02/03/2022 Número de solicitud: A123 tipo trabajador: Consultor Subtipo de trabajador: Independiente | Se presenta la sección crear solicitud Se presenta modal informativo La fecha de registro no está dentro de la vigencia establecida, por lo tanto la solicitud será rechazada ¿Estás seguro que deseas continuar? |
| 6 | Dar clic en el botón siguiente | |
| 7 | Dar clic en el boton continuar | Se presenta modal de rechazada "Has rechazado exitosamente la solicitud de vinculacion" |
| 8 | Dar clic en el boton aceptar | Se presenta la pantalla de vinculaciones |
| 9 | | |
| 10 | | |

Ilustración 140. DPC Rechazar Solicitud de Vinculación. Elaboración Propia.

Test

```

1 package com.heinsohn.crearVinculacionB0.test;
2
3 import ...
4 /**
5  * <b> Descripción:</b> Clase que permite ingresar los datos del stepper solicitud y la cual rechaza la solicitud
6  * <b> Caso de uso:</b> CUVI0364
7  *
8  * @author Danna Michelle Arevalo Hernandez <darevalo@heinsohn.com.co>
9  * @version
10 *
11 */
12 public class CrearVinculacionEsc01 {
13
14     private SeleniumTest selenium;
15     private LoginPage loginPage;
16     private MenuPage menuPage;
17     private CrearSolicitudPage crearSolicitudPage;
18
19 }

```

Ilustración 141. Test Crear Vinculación. Elaboración Propia.

```

29 /**
30 *
31 * Metodo encargado de implementar las instancias antes de la ejecucion de la prueba
32 *
33 * @author Jhonatan Morales <jhmoales@heinsohn.com.co>
34 *
35 */
36 @BeforeMethod(groups = { "crearSolicitudVinculacion" })
37 public void iniciarPrueba() {
38     // Se evalua que el objeto selenium no tenga instancias previas
39     if (selenium == null) {
40         selenium = new SeleniumTest(getClass().getSimpleName());
41         loginPage = new LoginPage(selenium.getDriver());
42         menuPage = new MenuPage(selenium.getDriver());
43         crearSolicitudPage = new CrearSolicitudPage(selenium.getDriver());
44     }
45 }

```

Ilustración 142. Test Crear Vinculación. Elaboración Propia.

```

47 *
48 * Metodo encargado de llamar al metodo de ingresar a la vinculacion y los datos del primer stepper "solicitud"
49 *
50 * @author Danna Michelle Arevalo Hernandez <darevalo@heinsohn.com.co>
51 *
52 */
53 @Test(groups = { "crearSolicitudVinculacion" })
54 public void crearVinculacionEsc01() {
55
56     // Ingresar a vinculaciones
57     loginPage.login(SeleniumProperties.getPropiedad(ConfigSeleniumPropiedades.USUARIO),
58         SeleniumProperties.getPropiedad(ConfigSeleniumPropiedades.CONTRASENA));
59     menuPage.clickOnMenu(menuPage.vinculaciones, menuPage.vinculaciones);
60
61     //Seleccionamos la card de crear vinculaciones
62     crearSolicitudPage.ingresarCardCrear();
63     ingresarDatosSolicitud(crearSolicitudPage);
64 }

```

Ilustración 143. Test Crear Vinculación. Elaboración Propia.

```

85 @ private void ingresarDatosSolicitud(CrearSolicitudPage crearSolicitudPage){
86     // Ingresan datos de la solicitud de vinculacion
87     CrearSolicitudDto crearSolicitudDto = new CrearSolicitudDto();
88     crearSolicitudDto.setCanal("Automático");
89     crearSolicitudDto.setFechaRegistro("01/03/2022");
90     crearSolicitudDto.setTipoTrabajador("Consultor");
91     crearSolicitudDto.setSubtipoTrabajador("Independiente");
92     String mensajeCanalAutomatico = crearSolicitudPage.verificarCanalAutomaticoNoEditable(crearSolicitudDto);
93     Assert.assertTrue( condition: true, message: "El campo numero de solicitud debe ser NO editable");
94     String mensajeCUA = crearSolicitudPage.verificarCUANoEditable(crearSolicitudDto);
95     Assert.assertTrue( condition: true, mensajeCUA);
96     crearSolicitudPage.ingresarDatosSolicitud(crearSolicitudDto);
97     String mensajeInformativo = crearSolicitudPage.validarMensajeInformativoFecha(crearSolicitudDto);
98     Assert.assertEquals(mensajeInformativo, expected: "La fecha de registro no está dentro de la vigencia establecida, por lo tanto",
99         message: "Se esperaba mensaje modal de informacion de la solicitud");
100     String mensajeRechazada = crearSolicitudPage.validarMensajeRechazoSol(crearSolicitudDto);
101     Assert.assertEquals(mensajeRechazada, expected: "Has rechazado exitosamente la solicitud de vinculación",
102         message: "Se esperaba mensaje modal de rechazada la solicitud");
103 }
104 /**

```

Ilustración 144. Test Crear Vinculación. Elaboración Propia.

```

84 /**
85  *
86  * Metodo encargado de cerrar la prueba
87  *
88  * @author Danna Michelle Arevalo Hernandez <darevalo@heinsohn.com.co>
89  *
90  */
91 @AfterMethod(groups = { "crearSolicitudVinculacion" })
92 public void cerrar() { selenium.getDriver().close(); }
95 }

```

Ilustración 145. Test Crear Vinculación. Elaboración Propia.

Page

```

11 public class CrearSolicitudPage {
12
13     private By cardCrear = By.id("cardCrear");
14     private By selectCanal = By.id("canal");
15     private By cua = By.id("cua");
16     private By numeroDeSolicitud = By.id("txtNumeroSolicitud");
17     private By fechaRegistro = By.id("fechaSolicitudInput");
18     private By selectTipoTrabajador = By.id("tipoAfiliado");
19     private By selectSubtipoTrabajador = By.id("subtipoTrabajador");
20     private By mensajeInformativo = By.id("idMensajeModal");
21     private By botonContinuar = By.id("btnContinuar");
22     private By botonCancelarMsn = By.id("btnCancelar");
23     private By botonAceptarMsnRechazada = By.id("btnAceptar");
24
25     private By botonSiguiente = By.id("stSiguiente");
26     private By botonCancelar = By.id("stCancelar");
27
28     private WebDriver driver;
29     private SeleniumWebDriverUtils utils;
30

```

Ilustración 146. Page Crear Vinculación. Elaboración Propia.

```

31
32     /**
33      * Constructor de la clase.
34      */
35     public CrearSolicitudPage(WebDriver driver) {
36         this.driver = driver;
37         utils = new SeleniumWebDriverUtils(this.driver);
38     }
39
40     /**
41      * Metodo encargado de entrar a la card de gestionar vinculacion
42      *
43      * @author Danna Michelle Arevalo Hernandez <darevalo@heinsohn.com.co>
44      */
45
46     public void ingresarCardCrear () {
47
48         utils.click(cardCrear);
49         utils.waitForLoading(Integer.parseInt(SeleniumProperties.getPropiedad(ConfigSeleniumPropiedades.TIME_OUT_DEFAULT)),
50             Integer.parseInt(SeleniumProperties.getPropiedad(ConfigSeleniumPropiedades.POLLING_DEFAULT)));
51     }
52

```

Ilustración 147. Método del Page Crear Vinculación. Elaboración Propia.

```

53 // **
54 *
55 * Método encargado de ingresar los datos de la vinculación
56 *
57 * @author Danna Michelle Arevalo Hernandez <darevalo@heinsohn.com.co>
58 *
59 */
60 @ public void ingresarDatosSolicitud(CrearSolicitudDto crearSolicitudDto){
61     utils.waitForOneSeconds();
62     utils.waitForOneSeconds();
63     utils.waitForOneSeconds();
64     utils.waitForOneSeconds();
65     // Seleccionamos el tipo de cua de la solicitud
66     if (crearSolicitudDto.getCanal() != null) {
67         utils.selectOptionList(selectCanal, crearSolicitudDto.getCanal());
68     }
69
70     utils.waitForOneSeconds();
71     utils.waitForOneSeconds();
72
73     if (crearSolicitudDto.getNumeroSolicitud() != null) {
74         utils.type(numeroDeSolicitud, crearSolicitudDto.getNumeroSolicitud());
75     }
76     utils.waitForOneSeconds();
77     utils.waitForOneSeconds();

```

Ilustración 148. Método del Page Crear Vinculación. Elaboración Propia.

```

101 *
102 * Método encargado de validar que el campo CUA sea no editable
103 *
104 * @author Danna Michelle Arevalo Hernandez <darevalo@heinsohn.com.co>
105 *
106 */
107 @ public String verificarCUANoEditable(CrearSolicitudDto crearSolicitudDto){
108     String mensaje= "";
109     if (crearSolicitudDto.getCua() != null) {
110         WebElement elemnt = utils.findElement(idElemento: "cua");
111         if (elemnt.isEnabled()){
112             mensaje ="El campo CUA debe ser no editable";
113         }
114     }
115     return mensaje;
116 }

```

Ilustración 149. Método del Page Crear Vinculación. Elaboración Propia.

```

117  /**
118  *
119  * Método encargado de validar que si el canal es igual a "automático" el
120  * numero de solicitud sea no editable
121  *
122  * @author Danna Michelle Arevalo Hernandez <darevalo@heinsohn.com.co>
123  *
124  */
125  @ public String verificarCanalAutomaticoNoEditable(CrearSolicitudDto crearSolicitudDto){
126      String mensaje= "";
127      if (crearSolicitudDto.getCanal() == "Automático") {
128          WebElement elemnt = utils.findElement( idElemento: "txtNumeroSolicitud");
129          if (elemnt.isEnabled()){
130              mensaje ="El campo numero de solicitud debe ser NO editable";
131          }
132      }
133      return mensaje;
134  }

```

Ilustración 150. Método del Page Crear Vinculación. Elaboración Propia.

```

135  /**
136  *
137  * Método encargado de validar que si el canal es igual a "preimpreso" el
138  * numero de solicitud sea editable
139  *
140  * @author Danna Michelle Arevalo Hernandez <darevalo@heinsohn.com.co>
141  *
142  */
143  @ public String verificarCanalPreimpresoEditable(CrearSolicitudDto crearSolicitudDto){
144      String mensaje= "";
145      if (crearSolicitudDto.getCanal() == "Preimpreso") {
146          WebElement elemnt = utils.findElement( idElemento: "txtNumeroSolicitud");
147          if (elemnt.isEnabled()){
148              mensaje ="El campo numero de solicitud debe ser editable";
149          }
150      }
151      return mensaje;
152  }

```

Ilustración 151. Método del Page Crear Vinculación. Elaboración Propia.

Dto

```

3 public class CrearSolicitudDto {
4     private String canal;
5     private String cua;
6     private String numeroSolicitud;
7     private String fechaRegistro;
8     private String tipoTrabajador;
9     private String subtipoTrabajador;
10
11     public String getCanal() { return canal; }
14
15     public void setCanal(String canal) { this.canal = canal; }
18
19     public String getCua() { return cua; }
22
23     public void setCua(String cua) { this.cua = cua; }
26
27     public String getNumeroSolicitud() { return numeroSolicitud; }
30
31     public void setNumeroSolicitud(String numeroSolicitud) { this.numeroSolicitud = numeroSolicitud; }
34
35     public String getFechaRegistro() { return fechaRegistro; }
38
39     public void setFechaRegistro(String fechaRegistro) { this.fechaRegistro = fechaRegistro; }
42

```

Ilustración 152. Dto Crear Vinculación. Elaboración Propia.

The screenshot displays the Selenium Test Results interface. At the top, it shows 'Test results' for '1 suite'. The suite is 'SuitePensionesObligatorias'. The results list shows two test cases, both of which passed:

- com.heinsohn.crearVinculacionB0.test.CrearVinculacionEsc01**: Passed. The test method is 'crearVinculacionEsc01'.
- com.heinsohn.crearVinculacionB0.test.CrearVinculacionEsc02**: Passed. The test method is 'crearVinculacionEsc02'.

The 'Info' section on the left provides details about the test suite:

- Path: D:\workspace\obligatorias\obligatorias\selenium-padre\vinculaciones-selenium\src\test\resources\testing.xml
- 4 tests
- 1 group
- Times
- Reporter output
- Ignored methods
- Chronological view

The 'Results' section indicates that 2 methods passed, 1 failed, and 1 passed (likely a typo for '1 failed, 1 passed'). The passed methods are 'crearVinculacionEsc02' and 'crearVinculacionEsc01'.

Ilustración 153. EPC Reporte Pruebas Vinculación de Asegurados. Elaboración Propia.

6.7. Funcionalidad Sucursales Empleador

Personas

En esta pantalla podrás consultar y gestionar personas según información personal, roles, relaciones, ubicaciones y cuentas bancarias. Teniendo una visión general de los procesos del asegurado.

[< VOLVER](#)

Consulta de datos


Tipo de identificación
Número de identificación tributaria ▼

Número de identificación: 2000222 Complemento CI


CUA


CREAR CONSULTAR

Información de persona

 **Banco de la republica**

Identificación **NIT 2000222**
Tipo persona **JURIDICA**


Roles


Sucursales

LA PAZ: MATRIZ ▼

Ilustración 154. Funcionalidad Sucursales Empleador. Elaboración Propia.


Roles


Sucursales

LA PAZ: MATRIZ ▼

LA PAZ: SUCURSAL ▼

PAJANCHA: SUCURSAL ▼

SANTA CRUZ DEL VALLE AMENO: SUCURSAL ▼

LA PAZ: SUCURSAL ▼

Ilustración 155. Funcionalidad Sucursales Empleador. Elaboración Propia.

LA PAZ: MATRIZ

Ubicaciones Información general Representante legal Información notaria

Dirección

Ciudad **LA PAZ** Provincia **MURILLO**

Departamento **LA PAZ** Zona **ZONA SUR**

Dirección **CRA 15-58-9** Número **78**

Referencia **referencia uno** Coordenada latitud (decimal)

Coordenada longitud (decimal)

Correo Electrónico

Correo Electrónico **amortiz@heinsohn.com**

Tipo de correo electrónico **INSTITUCIONAL**

Teléfono

Celular **32569748**

Teléfono

Extensión

Ilustración 156. Funcionalidad Sucursales Empleador. Elaboración Propia.

6.8. Funcionalidad Cierre de tramos laborales de entidades públicas y privadas

Buscar

Personas > Gestionar

Gestionar

En esta pantalla están agrupados los procesos para crear, consultar y gestionar toda la información de personas.

Personas

Realiza la gestión y consulta de la información de personas.

Empleador

Gestiona y consulta las solicitudes de los registros de empleador y consultor.

Asegurados

Gestiona y consulta las solicitudes de los registros del asegurado.

Cargue

Carga archivos que permiten ingresar solicitudes de asegurados.

Tramos laborales

Consulta y publica los procesos de ejecución en tramos laborales de entidades públicas y privadas.

Consultas

Filtra y presenta en detalle consultas de aseguramiento y tramos laborales.

Desarrollado por:

HEINSOHN
BUSINESS TECHNOLOGY

Ilustración 157. Funcionalidad Cierre de tramos laborales de entidades públicas y privadas. Elaboración Propia.

Tramos laborales

En esta pantalla están agrupados los procesos de ejecución de tramos laborales de entidades públicas y privadas.

[< VOLVER](#)

Fecha identificación ▼ Estado ▼ Fecha publicación ▼

| Fecha identificación | Estado | Fecha publicación | Medios de prensa | Número tramos laborales | Acciones |
|----------------------|----------|-------------------|------------------|-------------------------|---|
| 21/06/2022 | Generado | | | 2 |   |
| 20/04/2022 | Cerrado | | | 1 |   |

< 1 de 1 >

Ilustración 158. Funcionalidad Cierre de tramos laborales de entidades públicas y privadas. Elaboración Propia.

Esta funcionalidad permite identificar los tramos laborales para entidades públicas y privadas sin continuidad de aportes de acuerdo con la frecuencia y periodicidad parametrizada para el proceso.

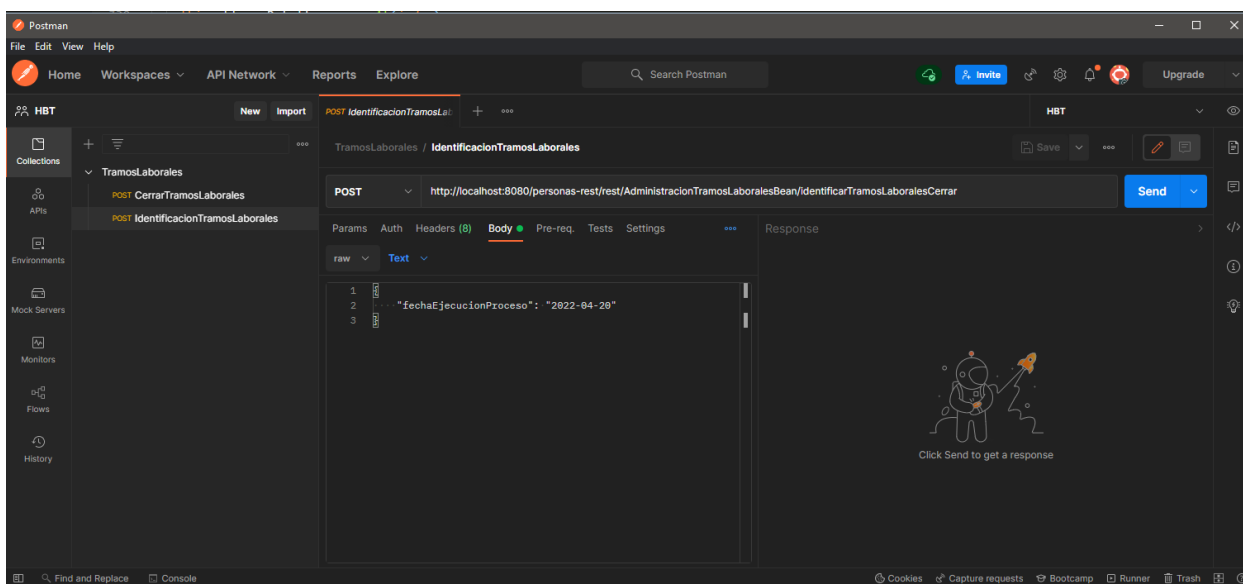


Ilustración 159. Ejecución Pruebas a través de Postman Funcionalidad Cierre de tramos laborales de entidades públicas y privadas. Elaboración Propia.

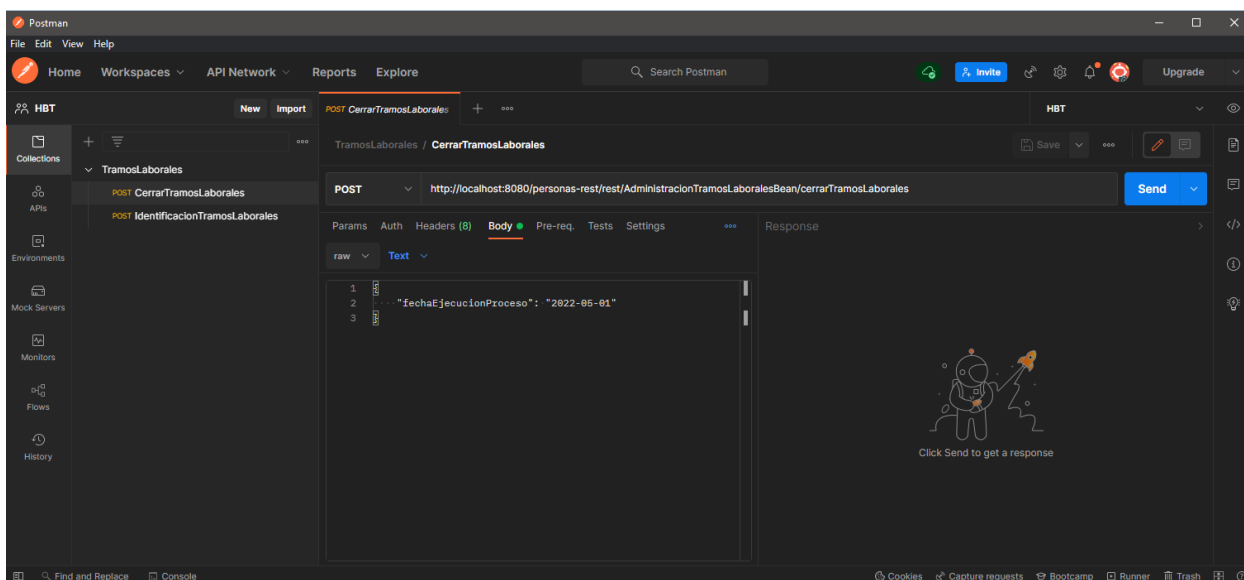


Ilustración 160. Ejecución Pruebas a través de Postman Funcionalidad Cierre de tramos laborales de entidades públicas y privadas. Elaboración Propia.

Esta funcionalidad permite cerrar los tramos laborales identificados previamente para entidades públicas y privadas sin continuidad de aportes de acuerdo con la frecuencia y periodicidad parametrizada para el proceso los casos de uso que contempla son los siguientes:

- CUPE1446 BOL Gestionar procesos tramos laborales entidades públicas y privadas
- CUPE1444 BOL Identificación de tramos laborales entidades públicas y privadas
- CUPE1445 BOL Cierre de tramos laborales entidades públicas y privadas
- CUPE0720 Actualizar entidades por cambio de nombre-16
- CUPE0719 Actualizar entidades por cambio de identificación-I30

6.9. Análisis de los resultados

Quiero hacer énfasis en que el objetivo de este proyecto era la implementación de pruebas automatizadas de las funcionalidades del módulo de personas dentro del aplicativo web de Pensiones en el ciclo de desarrollo de la empresa Heinsohn. Para cada funcionalidad se estudió los requerimientos solicitados por el cliente a través de los casos de uso, posteriormente se realizaron los escenarios y los diseños de pruebas que se iban a ejecutar por cada funcionalidad, se hizo la implementación del código para las pruebas, se utilizó la herramienta de Selenium junto con Java en las pruebas automatizadas para posteriormente ser ejecutadas y poder de esta forma evidenciar los reportes que generaron las pruebas ejecutadas. Al principio de este proyecto se había estimado un tiempo, pero a lo largo que se fue desarrollando las pruebas automatizadas con Selenium de las funcionalidades implementadas en el proyecto, sufrió unos cambios alargando el tiempo de estimación de un mes, pese a que sufrió cambios el tiempo estimado se cumplió con los requerimientos de este proyecto y el objetivo principal de este proyecto. Estas pruebas automatizadas ayudaron a detectar diferentes errores que se dieron a lo largo de cada funcionalidad dentro del aplicativo antes de ser entregado al cliente fueron corregidos y desplegados en la aplicación de la gestora de Bolivia. Se ha verificado que el proceso de pruebas automatizadas dentro del ciclo de desarrollo de la empresa se implementó positivamente. De igual forma estas pruebas automatizadas se ejecutan automáticamente continuamente dentro de la empresa para verificar que los procesos funcionen correctamente y se pueda detectar cualquier error si llega a sufrir algún cambio por modificaciones que se hagan en el aplicativo.

También hay que recalcar que hoy en día las pruebas automatizadas son muy importantes a la hora de hacer un aplicativo web, gracias a estas pruebas se puede entregar un producto más eficiente y mucho más confiable.

Capítulo 7. Referencias

- Rafael Cubas Montenegro. (s. f.). *Testing y Calidad de Software. Automatización de Pruebas con Selenium WebDriver*.
https://oa.upm.es/49320/1/PFC_RAFAEL_CUBAS_MONTENEGRO.pdf
- Adriano Tobías Vega Llobell. (2017). *Pruebas funcionales automatizadas para aplicaciones Web: Usando Selenium para aplicar pruebas de regresión automatizadas*.
<https://riunet.upv.es/bitstream/handle/10251/111170/Vega%20-%20Pruebas%20funcionales%20automatizadas%20para%20aplicaciones%20Web%3A%20usando%20Selenium%20para%20aplicar%20prue...pdf?sequence=1&isAllowed=y>
- *Selenium y la automatización de las pruebas*. (s. f.).
<https://www.juntadeandalucia.es/servicios/madeja/contenido/recurso/381>
- Víctor Manuel Soto Morales. (2018, abril). *4 herramientas de Selenium para crear tus pruebas automatizadas*. <https://www.pragma.com.co/blog/4-herramientas-de-selenium-para-crear-tus-pruebas-automatizadas>
- Matt Stine. (s. f.). *Automate User Acceptance Tests With Selenium 2.0*.
<https://dzone.com/refcardz/getting-started-selenium-20#:~:text=Selenium%202.0%20is%20a%20tool,project%20with%20the%20WebDriver%20project.>
- Selenium. (2022). *Selenium 2*.
https://www.selenium.dev/documentation/legacy/selenium_2/
- Angela Paola Duquino Sánchez. (2020). *Automatización de un Sistema de Pruebas de Software para la Optimización del Proceso de Calidad de DetectIDTM*.

<https://repositorio.unal.edu.co/bitstream/handle/unal/79288/Tesis%20Maestri%CC%81a%20Angela%20Paola%20Duquino%20S.pdf?sequence=1&isAllowed=y>

- Gundecha, U. (2019). *SELENIUM WEBDRIVER 3 PRACTICAL GUIDE - SECOND EDITION* (2.^a ed.).
- CRISTIAN ALEJANDRO RIVERA MARTÍNEZ. (2018). *AUTOMATIZACIÓN DE PRUEBAS DE REGRESIÓN*.
<https://repositorio.uchile.cl/bitstream/handle/2250/165608/Automatizaci%C3%B3n-de-pruebas-de-regresi%C3%B3n.pdf?sequence=1&isAllowed=y>
- MIGUEL ÁNGEL DE DIOS. (2022, mayo). *Scrum: qué es y cómo funciona este marco de trabajo*. <https://www.wearemarketing.com/es/blog/metodologia-scrum-que-es-y-como-funciona.html>

Capítulo 8. Trabajo futuro

Automatización de aplicaciones en dispositivos o entornos móviles, las herramientas basadas en Selenium WebDriver para dispositivos móviles como Android, iPhone, iOS, esto nos ayudaría a ver muchos más fallos que no se dan en versiones de escritorio.