

ANEXO TÉCNICO

**PRESTACIÓN DE LOS SERVICIOS DE DESARROLLO DE SOFTWARE CON
METODOLOGÍAS AGILES PARA LA CONSTRUCCIÓN DE APLICACIONES WEB DEL
ICFES, CON LAS ESPECIFICACIONES TÉCNICAS CONTENIDAS EN EL ANEXO
TÉCNICO.**

CONTENIDO

INTRODUCCIÓN	3
1.1. OBJETO A CONTRATAR	3
1.2. CARACTERÍSTICAS DEL SERVICIO	3
1.3. OBLIGACIONES DEL CONTRATISTA:	4
1.4. RESPONSABILIDADES DE LOS RECURSOS REQUERIDOS:	4
1.5. AMBIENTES Y DESPLIEGUES ENTRE AMBIENTES	5
1.6. GARANTÍA DE LOS PRODUCTOS	6
1.7. TRANSFERENCIA DE CONOCIMIENTO	6
1.8. LINEAMIENTOS Y BUENAS PRÁCTICAS	6
EQUIPO DE TRABAJO	19
CONDICIONES GENERALES	19

INTRODUCCIÓN

Alineados con el PEI y buscando dar cumplimiento al objetivo No. 8 del mapa estratégico: Fortalecer el uso de la tecnología, que es transversal a todos los ejes estratégicos, además de soportar y facilitar el cumplimiento general del Plan estratégico Institucional, pues provee las herramientas tecnológicas que permitirán su implementación efectiva en el Decreto 1078 de 2015 en su artículo 2.2.9.1.2.1, Gobierno en Línea hace mención a los componentes establecidos para que las entidades se alineen a la estrategia, de los cuales dos ejes temáticos son TIC PARA SERVICIOS y TIC PARA LA GESTIÓN donde se resalta la importancia del aprovisionamiento de trámites y servicios a través de medios electrónicos y la necesidad de dar solución a ciudadanos y entidades en condiciones de calidad, facilidad de uso, mejoramiento continuo y el uso estratégico de la tecnología para hacer más eficaz la gestión administrativa.

Con el objetivo de mantener el mejoramiento continuo de la plataforma tecnológica y la constante evolución y ampliación de los servicios que se ofrece a través de los sistemas de información de la entidad, la Subdirección de Desarrollo de Aplicaciones requiere contratar los servicios de fábrica de desarrollo de software con metodologías ágiles para aplicaciones web.

1.1. OBJETO A CONTRATAR

Prestar los servicios de fábrica de desarrollo de software con metodologías ágiles para aplicaciones web.

1.2. CARACTERÍSTICAS DEL SERVICIO

El alcance incluye construir las funcionalidades para los sistemas de información que solicite el ICFES cumpliendo con los lineamientos técnicos definidos por la Dirección de Tecnología.

A continuación, se describen las condiciones y actividades mínimas requeridas que el contratista deberá realizar en cumplimiento del objeto del contrato:

1.3. OBLIGACIONES DEL CONTRATISTA:

1. Proveer los recursos solicitados según los perfiles definidos (ver sección 2. PERFILLES.) por la entidad en condiciones de tiempo y costo pactados.
2. Contar con la infraestructura tecnológica, física y administrativa necesaria para la ejecución del contrato.
3. Cumplir con la transferencia de conocimiento sobre los productos construidos al supervisor del contrato o a quien el designe para tal fin.
4. Cumplir con la garantía de cada uno de los desarrollos de acuerdo con los requerimientos establecidos en el numeral 1.6 del anexo técnico.

Las anteriores obligaciones, además de las dispuestas en el pliego de condiciones, adendas y la minuta contractual.

1.4. RESPONSABILIDADES DE LOS RECURSOS REQUERIDOS:

1. Construir las funcionalidades solicitadas para los sistemas de información de la entidad.
2. Seguir la metodología de proyectos de desarrollo y/o mantenimiento de software, definida por el ICFES.
3. Apoyar el diseño de las historias de usuario a su cargo usando las herramientas definidas por el ICFES y siguiendo los lineamientos definidos por la Dirección de Tecnología.
4. Apoyar la implementación de las historias de usuario asignadas, de acuerdo con los lineamientos de la metodología de desarrollo definida por el ICFES.
5. Apoyar la ejecución de los planes de pruebas unitarias y funcionales que garanticen la operación correcta de las historias de usuario implementadas.
6. Documentar y entregar las historias de usuario implementadas, el manual técnico y de usuario final, detectar fallas de desempeño y seguridad, y recomendar las soluciones y/o estrategias a estas eventualidades.
7. Apoyar la entrega y capacitación de los aplicativos a su cargo, a la persona que el ICFES designe para esta labor.
8. Apoyar la asesoría y capacitación a los funcionarios del Instituto, según se indique.
9. Entregar el código fuente de los nuevos aplicativos o mantenimientos realizados.
10. Apoyar en la debida sincronización del código fuente en el repositorio definido para tal fin.
11. Garantizar la calidad de cada historia de usuario asignada y realizar acompañamiento en la estabilización.

12. Apoyar en el análisis de los requerimientos que sean necesarios usando historias de usuario y/o modificar las ya existentes, de acuerdo a las necesidades del Instituto.
13. Apoyar la ejecución de las pruebas unitarias periódicas según definición de la metodología para los componentes de software que construya y entregar informes con los resultados al líder técnico.
14. Asistir a las reuniones diarias del Sprint con su grupo de trabajo.
15. Reportar las actividades ejecutadas en la herramienta que para tal fin asigne el ICFES.
16. Asistir a las reuniones que se programen, relacionadas con el objeto del contrato.
17. Apoyar las actividades relacionadas con la ejecución de pruebas, puesta en producción y estabilización.
18. Respetar los lineamientos de buenas prácticas de desarrollo establecidas por el arquitecto y proponer ideas que permitan mejorar el desarrollo y la implementación de los componentes.
19. Realizar la documentación y entregar para las historias de usuario implementadas, el manual técnico y de usuario final.
20. Mantener correctamente en el repositorio del ICFES según las prácticas definidas por la Dirección de Tecnología de la entidad el código fuente de los nuevos desarrollos o mantenimientos realizados.
21. Guardar a favor del ICFES el sigilo y la reserva de la información que se le entregue, obtenga y manipule en razón del contrato.
22. Mantener en forma confidencial, todos los datos e informaciones a las cuales tenga acceso siendo esta confidencialidad continua y sin vencimiento ni por terminación, ni por la declaratoria de caducidad de la orden de servicios.

1.5. AMBIENTES Y DESPLIEGUES ENTRE AMBIENTES

La responsabilidad de gestión y aprovisionamiento de los ambientes estará a cargo de:

Contratista Seleccionado

- Ambientes de Desarrollo

Contratista Seleccionado

- Ambientes de Pruebas de Calidad
- Ambientes de Pruebas UAT
- Ambientes de Producción

Los despliegues entre ambientes de desarrollo, aseguramiento de calidad, pruebas UAT y producción los realizarán en conjunto los equipos técnicos del contratista y el contratante, los despliegues considerados son:

- Pasos de desarrollo a pruebas de calidad

- Pasos de pruebas de calidad a pruebas UAT
- Pasos de pruebas UAT a producción

Todos los despliegues se realizarán en las instalaciones del contratante.

1.6. GARANTÍA DE LOS PRODUCTOS

Los desarrollos ejecutados por el contratista de acuerdo con el cronograma definido, deben ser avalados por el supervisor del contrato y los mismos tendrán un periodo de garantía de sesenta (60) días calendario contados a partir de la aprobación señalada.

1.7. TRANSFERENCIA DE CONOCIMIENTO

La transferencia de conocimiento que realizará el contratista deberá garantizar los siguientes aspectos:

- Que el contratante obtenga el conocimiento explícito adquirido por el contratista durante la construcción de los productos generados en la ejecución del contrato.
- La transferencia cubrirá aspectos funcionales y técnicos.
- Los recursos utilizados para la transferencia de conocimiento pueden ser documentos, medios audiovisuales o cualquier otro material que garantice la adecuada transferencia.
- La transferencia de conocimiento está dirigido a los líderes funcionales y técnicos que designe el supervisor del contrato.
- La transferencia de conocimiento se realizará en las instalaciones del ICFES.

1.8. LINEAMIENTOS Y BUENAS PRÁCTICAS

Todos los productos de desarrollo construidos deben cumplir los siguientes lineamientos y buenas prácticas:

1.8.1. Documentación de proyectos

#	Documentación por Proyecto	Elaboración	Responsable
0	Arquitectura de referencia.	Líder de Arquitectura	Entidad
1	Documentos de procesos.	Planeación / Procesos	Entidad

2	Documentos de diseño	Líder Desarrollo/ Desarrollador	Contratista
	2.1 Diagrama de clases por proceso	Líder Desarrollo/ Desarrollador	Contratista
	2.2 Diagrama de clases (si aplica) y secuencia por cada caso de uso/historia	Líder Desarrollo/ Desarrollador	Contratista
3	Documentos de casos de uso/historia	Analistas funcionales	Entidad
4	Documentos casos de pruebas.	Pruebas funcionales	Contratista - Entidad
5	Diseño conceptual de bases de datos	Arquitectos/Analistas	Entidad
	5.1 Modelo conceptual de bases de datos	Arquitectos/Analistas	Entidad
	5.2 Diccionario conceptual de bases de datos	Arquitectos/Analistas	Entidad
6	Diseño físico de datos	Arquitectos/Analistas	Entidad
	6.1 Modelo de datos físico por proceso	Arquitectos/Analistas	Entidad
	6.2 Diccionario de datos	Arquitectos/Analistas	Entidad
	6.2.1 Documentación de tablas, campos, secuencias, opcionalidad	Arquitectos/Analistas	Entidad
	6.2.2 Documentación de restricciones (primarias, foráneas, rangos, valores)	Arquitectos/Analistas	Entidad
	6.2.3 Documentación de vistas	Arquitectos/Analistas	Entidad

	6.2.4 Documentación de procesos (paquetes, procedimientos, funciones, disparadores).	Arquitectos/Analistas	Entidad
	6.2.5 Anexos (scripts o referenciar la ubicación).	Arquitectos/Analistas	Entidad
7	Especificaciones suplementarias por Caso de uso (que aplique)	Líder Desarrollo	Contratista
8	Documento de arquitectura	Arquitectos/Líder Desarrollo	Contratista - Entidad
	8.1 Vista de Casos de Uso	Arquitectos/Líder Desarrollo	Contratista - Entidad
	8.2 Matriz Tecnológica	Arquitectos/Líder Desarrollo	Contratista - Entidad
	8.3 Vista Lógica	Arquitectos/Líder Desarrollo	Contratista - Entidad
	8.4 Vista de Procesos	Arquitectos/Líder Desarrollo	Contratista - Entidad
	8.5 Vista de Despliegue	Arquitectos/Líder Desarrollo	Contratista - Entidad
	8.6 Vista de Datos	Arquitectos/Líder Desarrollo	Contratista - Entidad
	8.7 Componentes Transversales	Arquitectos/Líder Desarrollo	Contratista - Entidad
	8.8 Calidad	Arquitectos/Líder Desarrollo	Contratista - Entidad
9	Reporte final de pruebas funcionales: incluyendo cantidad de casos de prueba, listado de hallazgos, hallazgos abiertos, etc.	Scrum Team (Líder Desarrollo/Scrum Master, Desarrolladores, Analistas de Pruebas, Analistas Funcionales)	Contratista - Entidad
10	Reporte de pruebas no funcionales - solicitadas	Scrum Team (Líder Desarrollo/Scrum Master, Desarrolladores, Analistas de Pruebas, Analistas Funcionales)	Contratista - Entidad
	10.1 - Reporte final de pruebas de cada tipo realizado (carga, stress, picos , etc)	Scrum Team (Líder Desarrollo/Scrum Master, Desarrolladores, Analistas de Pruebas, Analistas Funcionales)	Contratista - Entidad

	10.2 - Reporte final de pruebas de seguridad	Scrum Team (Líder Desarrollo/Scrum Master, Desarrolladores, Analistas de Pruebas, Analistas Funcionales)	Contratista - Entidad
11	Implementación	Líder Desarrollo/ Desarrollador	Contratista - Entidad
	11.1 Código fuente	Líder Desarrollo/ Desarrollador	Contratista
	11.2 Documentación Api Clases (Java Doc)	Líder Desarrollo/ Desarrollador	Contratista
	11.3 Scripts	Arquitectos/ Líder Desarrollo/ Desarrollador	Contratista - Entidad
12	Manuales de usuario.	Procesos	Contratista
13	Manuales de técnico	Líder Desarrollo/ Desarrollador	Contratista
14	Actas de las capacitaciones técnicas recibidas: Puntos 0 al 15.	Scrum Team (Líder Desarrollo/Scrum Master, Desarrolladores, Analistas de Pruebas, Analistas Funcionales), Gerente de Proyecto	Contratista - Entidad
15	Acta de entrega del aplicativo, en la cual se adjunte toda la documentación anterior, el nombre de la persona del nuevo sistema misional que se encargará de la estabilización, y la fecha en la que finalizará la estabilización (Normalmente un ciclo de operación).	Scrum Team (Líder Desarrollo/Scrum Master, Desarrolladores, Analistas de Pruebas, Analistas Funcionales) , Gerente de Proyecto	Contratista - Entidad

1.8.2. Estructura de Proyectos y Paquetes son definidas por el Arquitecto

Nomenclatura de Paquetes

Proyecto EJB

- co.gov.icfes.misional.<subsistema>.<modulo|comun>.ejb.servicio
- co.gov.icfes.misional.<subsistema>.ejb.fachada
- co.gov.icfes.misional.<subsistema>.interceptores

Proyecto WEB

- co.gov.icfes.misional.<subsistema>.<modulo|comun>.web.bean (La premisa es que cada página tiene su propio backing bean y se llaman igual)
- WebContent/pages/<modulo|comun>/<pagina>.xhtml

- WebContent/resources/<css|formatos|images|js>

Ejemplo

- ▲ citacion-ejb
 - ▲ ejbModule
 - co.gov.icfes.misional.citacion.comun.ejb.servicio
 - co.gov.icfes.misional.citacion.configuracion.ejb.servicio
 - co.gov.icfes.misional.citacion.ejb.fachada
 - co.gov.icfes.misional.citacion.gestion.ejb.servicio
 - co.gov.icfes.misional.citacion.seguimiento.ejb.servicio
 - META-INF
 - ▶ JRE System Library [JavaSE-1.7]
 - lib
-
- ▲ citacion-web
 - ▶ JAX-WS Web Services
 - ▶ Deployment Descriptor: citacion-web
 - ▲ Java Resources
 - ▲ src
 - co.gov.icfes.misional.citacion.comun.web.bean
 - co.gov.icfes.misional.citacion.configuracion.web.bean
 - co.gov.icfes.misional.citacion.gestion.web.bean
 - co.gov.icfes.misional.citacion.seguimiento.web.bean
 - ▶ Libraries
 - ▶ JavaScript Resources
 - ▶ build
 - ▲ WebContent
 - ▶ META-INF
 - ▲ pages
 - ▲ citacion
 - comun
 - configuracion
 - gestion
 - seguimiento
 - comun
 - ▲ resources
 - css
 - images
 - ▶ WEB-INF
 - index.jsp

1.8.3. Definición de Pantallas

El contratista seleccionado debe cumplir con lo definido en el documento **Lineamientos gráficos de usabilidad y accesibilidad ICFES TI.doc** adjunto en este proceso.

1.8.4. Ejecución de PMD y checkstyle

Cada funcionalidad que se desarrolle debemos realizar la entrega de los diferentes artefactos (diseño, código, pruebas unitarias, revisión de código), en el anterior lineamiento se explicó los artefactos de diseño, en este lineamiento veremos pruebas unitarias y revisión de código estático.

Ejecución PMD y Checkstyle

Tenemos la definición de nuestras diferentes herramientas y plugins (eclipse, pmd, checkstyle, cobertura, svn, mantis, otros), así como el uso, configuración y reglas de revisión.

La **Figura 1** nos muestra la estructura del reporte html generado donde inicialmente tenemos el listado de archivos analizados, la cantidad de reglas violadas discriminado por severidad. Posteriormente tenemos mayor información de las reglas violadas para luego tener un cruce de reglas por clases.


Reporte Revisión Código Estático					
(2015-08-21-05:00; 00:20:25.582-05:00)					
					
Total Archivos					
archivo	high	medium	low	total	
co.gov.icfes.misional.administracion.autenticacion.helper.PanelControlAgrupadorHelper.java	2	26	12	40	
co.gov.icfes.util_.java	0	0	1	1	
co.gov.icfes.util.VariablesConfiguracionSistema.java	0	11	0	11	
	2	37	13	52	
Total Reglas					
categoria	herramienta	prioridad	regla	ocurrencias	
controversial					
	pmd	3	UnnecessaryConstructor	2	
	pmd	3	CallSuperInConstructor	2	
design					
				52	
Detalles por clase					
co.gov.icfes.misional.administracion.autenticacion.helper.PanelControlAgrupadorHelper.java					
tool	priority	line	category	rule	message
pmd	1	16	design	EmptyMethodInAbstractClassShouldBeAbstract	EmptyMethodInAbstractClassShouldBeAbstract
pmd	1	16	design	EmptyMethodInAbstractClassShouldBeAbstract	EmptyMethodInAbstractClassShouldBeAbstract
checkstyle	2	0	style	Missing package-info.java file.	warning

Figura 1. Estructura reporte generado

Conjunto de reglas a verificar:

- AbstractClassName
- AvoidNestedBlocks
- AvoidStarImport
- ConstantName
- CyclomaticComplexity
- DeclarationOrder
- DefaultComesLast
- EmptyBlock
- EmptyCatchBlock
- EmptyForInitializerPad
- EmptyForIteratorPad
- EmptyStatement
- EqualsAvoidNull
- EqualsHashCode
- FileLength
- FinalClass
- FinalLocalVariable
- FinalParameters
- IllegalCatch
- IllegalImport
- IllegalInstantiation
- IllegalThrows
- InnerAssignment
- Indentation
- InnerAssignment
- JavadocMethod
- JavadocType
- JavadocVariable
- JavadocStyle
- LineLength
- LocalFinalVariableName
- LocalVariableName
- MagicNumber
- MemberName
- MethodName
- MethodCount
- MethodLength
- MissingSwitchDefault
- ModifierOrder
- NeedBraces
- PackageName
- ParameterName
- ParameterNumber
- RedundantImport
- StaticVariableName
- SimplifyBooleanExpression
- SimplifyBooleanReturn

- TodoComment
- TypeName
- UnusedImports
- UpperEll

Entrega Sprint y Modulo

El líder de desarrollo se encarga de:

- Configuración y versionamiento de ant (revisioCodigo.xml) con el que se genera el reporte para el modulo, el cual contiene todas la clases creadas y modificadas desde el sprint 0. La estructura de nombrado del archivo es: 'Modulo'.xml
- Versionamiento de reporte HTML de revisión de código del módulo. La estructura de nombrado del archivo es: 'Modulo'_reporte_pmd_checkstyle.html
- Versionamiento de reporte XML de revisión de código del módulo. La estructura de nombrado del archivo es: 'Modulo'_reporte_pmd_checkstyle.xml

1.8.5. Ejecución de pruebas unitarias

Introducción

El presente manual es una guía de iniciación en el manejo de JSFUnit. El documento se estructura en varios apartados, en los que se han incluido los principales aspectos que debe saber un usuario novato de JSFUnit para empezar a trabajar con la herramienta.

JSF Unit es un framework de pruebas para aplicaciones JSF. Está diseñada para permitir una completa integración de pruebas y pruebas de unidad de las aplicaciones JSF utilizando un API simplificado. JSFUnit corre dentro de un contenedor, el cual provee al desarrollador completo acceso al control de los Beans, el FacesContext, expresiones El y al componente interno JSF Tree. Al mismo tiempo tiene acceso al HTML de salida enviado para cada uno de los request del cliente. <http://jsfunit.jboss.org/>

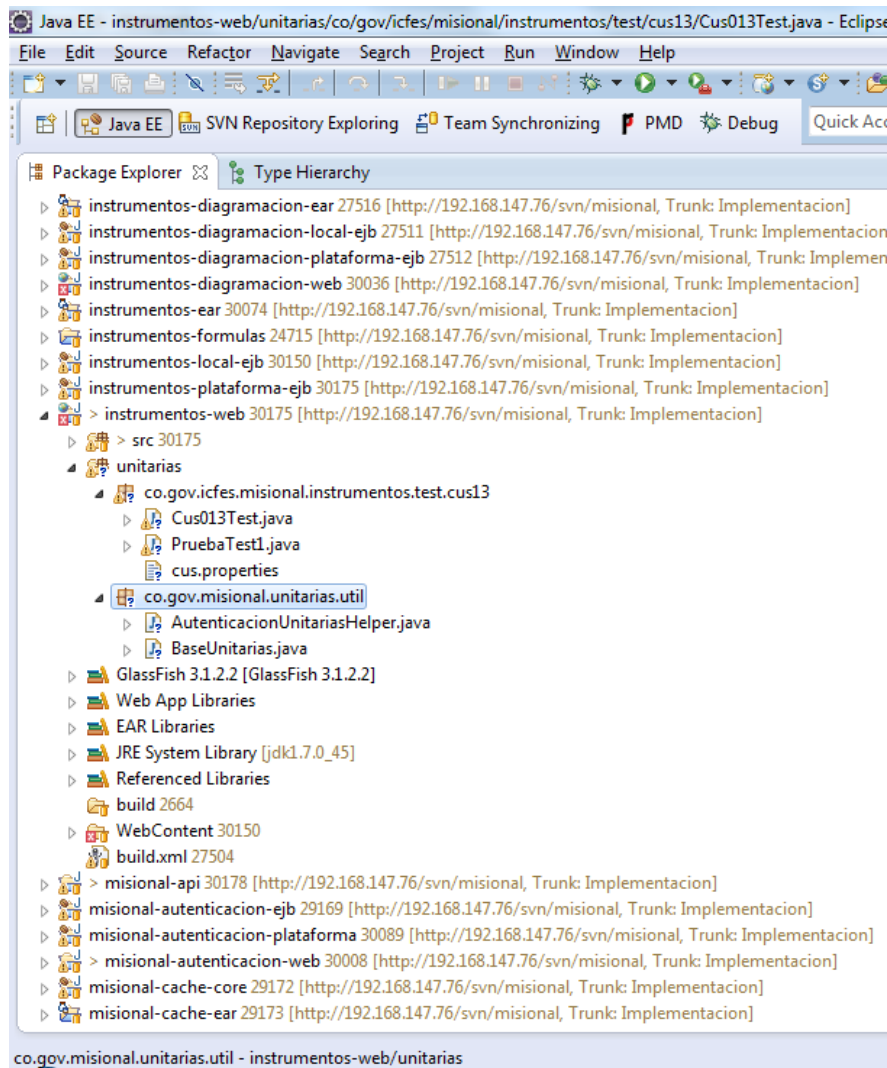
JUnit es un conjunto de clases (framework) que permite realizar la ejecución de clases Java de manera controlada, para poder evaluar si el funcionamiento de cada uno de los métodos de la clase se comporta como se espera. Es decir, en función de algún valor de entrada se evalúa el valor de retorno esperado; si la clase cumple con la especificación, entonces JUnit devolverá que el método de la clase pasó exitosamente la prueba; en caso de que el valor esperado sea diferente al que regresó el método durante la ejecución, JUnit devolverá un fallo en el método correspondiente. <http://junit.org/>

HtmlUnit es un navegador web sin interfaz de usuario escrito en Java. Permite la manipulación de alto nivel de sitios web a partir de otro código Java, incluyendo el llenado y envío de formularios y el hacer click en hiperenlaces. También proporciona acceso a la estructura y los detalles de las páginas web recibidas. HtmlUnit emula el comportamiento de las partes del navegador, incluyendo los aspectos de bajo nivel de los protocolos TCP/IP y HTTP. Una secuencia como `getPage(url)`, `getLinkWith("Click here")`, `click()` permite al usuario navegar a través de hipertexto y obtener páginas web, incluyendo HTML, JavaScript, AJAX y cookies. <http://htmlunit.sourceforge.net/>

XPath (XML Path Language) es un lenguaje que permite construir expresiones que recorren y procesan un documento XML. La idea es parecida a las expresiones regulares para seleccionar partes de un texto sin atributos (plain text). XPath permite buscar y seleccionar teniendo en cuenta la estructura jerárquica del XML. XPath fue creado para su uso en el estándar XSLT, en el que se usa para seleccionar y examinar la estructura del documento de entrada de la transformación. http://www.w3schools.com/xsl/xpath_syntax.asp

Estructura de paquetes

En el *source* de cada módulo web se debe tener un *source* llamado **unitarias** la cual permite desacoplar las implementaciones de los test del código fuente de negocio.



1. En el *source* llamado **unitarias** se crearan los test de pruebas unitarias donde se creara un paquete `co.gov.icfes.misional.MODULO.test.NUMERO_HISTORIA_DE_USUARIO`.
2. Dentro de este paquete se creará una clase con el identificador `CUS_NUMERO_HISTORIA_DE_USUARIO_Test`.
3. En el paquete del caso de uso del *source* **unitarias** del proyecto de crear un archivo `CUS_NUMERO_HISTORIA_DE_USUARIO.properties` con el fin de tener los set de pruebas que se van a ejecutar.

La Clase **AutenticacionUnitariasHelper** es la encargada de comunicar el Modulo con el SSO y ademas es el encargado de iniciar sesión y otorgarnos un TOKEN.

La Clase **BaseUnitarias** es la encargada de tener la lógica para obtener los set de prueba contenidos en el archivo `CUS_NUMERO_HISTORIA_DE_USUARIO.properties`

1.8.6. Checklist entrega desarrollo

En el formato ChecklistEntregaArquitectura.xlsx está el ckecklist de los artefactos y actividades solicitados por arquitectura para entrega del desarrollo de las funcionalidades.

La **figura 1** nos muestra una imagen del formato donde primero colocamos el nombre del proyecto y el sprint que estamos entregando, luego tenemos enumerados los registros de fechas de las operaciones: entregado, revisión, completado y aceptado según sea el caso; en el siguiente cuadro tenemos la lista artefactos y actividades y por último el cuadro de historias o casos de uso con los check de entrega y verificación. Como se ve en el ejemplo tenemos 4 operaciones.

PROYECTO/MODULO: Citación		SPRINT: Uno
---------------------------	--	-------------

N°	FECHA REGISTRO	DESCRIPCIÓN	ESTADO	RESPONSABLE
1	01/02/2016	Entrega Sprint	Entregado	Líder
2	03/02/2016	Validación Entrega	Incompleto	Arquitectura
3	10/02/2016	Completar Entrega	Completado	Líder
4	12/02/2016	Verificación Entrega	Aceptado	Arquitectura

Desarrollador/Líder	Revisión PMD	PMD
Desarrollador/Líder	Diseños	DIS
Desarrollador/Líder	Revisión CheckStyle	CHK
Desarrollador/Líder	Revisión Pruebas Unitarias	PU
Arquitectura Datos/Desarrollador/Líder	Revisión de Query	SQL
Desarrollador/Líder	Revisión de Pool (1 Conexión)	POOL
Arquitectura Datos/Desarrollador/Líder	Revisión Smoke Test	SMK
Arquitectura Datos/Desarrollador/Líder	Certificación Base de Datos (Esquema pruebas no funcionales)	CBD

Caso de Uso N°	PMD				CHK				PU				SQL				POOL				SMK				CBD							
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4				
CUS XXX - HIST XXX	✓	✓			✓	✓			✓	✓			✓	✓			✓	✓			✓	✓			✓	✓			✓	✓		
CUS XXX - HIST XXX	✓	x	✓	✓	✓	x	✓	✓	✓	✓			✓	x	✓	✓	✓	x	✓	✓	✓	✓			✓	x	✓	✓	✓	✓		

Figura 1

Ejemplo llenado Nombre HU

Identificación	Caso de Uso/Historia	Descripción
	N°	
010	Ajuste de validación de calendario, para los procesos de aplicación se debe validar el calendario de la sede jornada y no del establecimiento	
011	Realizar ajustes respectivo para cuando se restringe el acceso a un agrupador por calendario para ciertos exámenes.	

Veamos la traza de llenado de este ejemplo. En caso de que el ítem no aplique colocamos **-1 = N/A**

- 1 = No aplica
- 0 -99= No cumplido
- 100 = Cumplido

Caso de Uso		DIS				PMD				CHK			
		1	2	3	4	1	2	3	4	1	2	3	4
091	Como Administrador de evaluación Yo puedo ejecutar el ordenamiento con la cantidad de citados en un salón Para asignar un orden de puesto consecutivo a inscritos	●				✓				✓			
092	Como Administrador de Evaluación Yo puedo realizar una segunda numeración Para asignar un consecutivo a inscritos adicionales.					✗			✗				

1. El día 01/02/2016 el líder lleno el formato e hizo entrega del sprint y diligencio la respectiva columna de entrega de los artefactos

Caso de Uso	PMD	CHK	PU	SQL	POOL	SMK	CBD
N°	1	1	1	1	1	1	1
CUS XXX - HIST XXX	✓	✓	✓	✓	✓	✓	✓
CUS XXX - HIST XXX	✓	✓	✓	✓	✓	✓	✓

2. El día 03/02/2016 arquitectura hace la revisión de la entrega y llena la respectiva columna de revisión de los artefactos según lista de registro de fechas

Caso de Uso	PMD		CHK		PU		SQL		POOL		SMK		CBD	
N°	1	2	1	2	1	2	1	2	1	2	1	2	1	2
CUS XXX - HIST XXX	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗
CUS XXX - HIST XXX	✓	✗	✓	✗	✓	✓	✓	✗	✓	✗	✓	✓	✓	✗

3. El día 10/02/2016 el líder revisa el checklist, realiza los ajustes solicitados y diligencia la respectiva columna de la nueva entrega de los artefactos según lista de registro de fechas

Caso de Uso	PMD			CHK			PU			SQL			POOL			SMK			CBD		
N°	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3
CUS XXX - HIST XXX	✓	✓		✓	✓		✓	✓		✓	✓		✓	✓		✓	✓		✓	✗	✓
CUS XXX - HIST XXX	✓	✗	✓	✓	✗	✓	✓	✓	✓	✓	✗	✓	✓	✓	✓	✓	✓	✓	✓	✗	✓

4. El día 12/02/2016 arquitectura hace la validación de la nueva entrega y llena la respectiva columna de revisión de los artefactos según lista de registro de fechas

Caso de Uso N°	PMD				CHK				PU				SQL				POOL				SMK				CBD							
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4				
CUS XXX - HIST XXX	✓	✓			✓	✓			✓	✓			✓	✓			✓	✓			✓	✓			✓	✓			✓	*	✓	✓
CUS XXX - HIST XXX	✓	*	✓	✓	✓	*	✓	✓	✓	✓			✓	*	✓	✓	✓	*	✓	✓	✓	✓			✓	✓			✓	*	✓	✓

1.8.7. Administrar adecuadamente recursos costosos (Acceso BD, Acceso Disco, Memoria)

Será necesario validar en cada implementación de un producto de software con el grupo de Arquitectura de la entidad los siguientes puntos:

- Uso de cache cuando se requiera información que se mantiene estable en el tiempo.
- Optimizar el rendimiento consultar SQL
- Evaluar la posibilidad de consultas o acceso en batch vs accesos individuales.

EQUIPO DE TRABAJO

Para la ejecución del contrato el proponente debe ofrecer el equipo de trabajo que se especifica en el presente numeral:

ROL	REQUISITOS DEL PERFIL	
OCHO (08) ANALISTAS DE DESARROLLO	Formación	Profesional en Ingeniería de Sistemas o Ingeniería Electrónica o Ingeniería de Software, Telecomunicaciones y Telemática.
	Experiencia	<p>Dos (02) años de experiencia en implementación de proyectos con todas tecnologías que se relacionan a continuación:</p> <p>Arquitectura JEE y/o Hexagonal, Desarrollo de aplicaciones WEB (responsive), Desarrollo Java y aplicaciones JEE, diseño de bases de datos y SQL, Base de datos Oracle y PostgreSQL.</p> <p>Participación en la implementación de un (1) proyecto usando alguna de las siguientes tecnologías: Angular, y/o TypeScript y/o JavaScript y/o HTML5 y/o CSS3.</p>
	Certificaciones	Java Programmer y/o Scrum Master (metodologías ágiles)

Nota. El tiempo de experiencia para el equipo de trabajo no puede trasladarse.

CONDICIONES GENERALES

A continuación, se establecen las condiciones generales de la prestación del servicio:

1. Algunas actividades podrán ser ejecutadas en las instalaciones del ICFES, ubicada en la ciudad de Bogotá en la Calle 26 # 69 – 76 Piso 16 o en la sede del contratista.
2. El ICFES conocerá y aprobará las hojas de vida de los perfiles solicitados.
3. Versiones y Herramientas necesarias para la ejecución del servicio:
 - a. Repositorios, integración y revisión continua:
 - i. SVN y GIT (Repositorio de información)
 - ii. JENKINS (Integración Continua)
 - iii. PMD, CHECKSTYLE (Revisión continua)
 - b. Versiones de Java e IDE para el desarrollo:
 - i. JAVA 7, JAVA 8
 - ii. ECLIPSE NEON, ECLIPSE OXYGEN
 - c. Versiones de Base de Datos:
 - i. ORACLE 11G, ORACLE 12C

- ii. PostgreSQL 9.6.3
- d. Uso de las Bases de Datos: Las bases de datos serán usadas únicamente como repositorio de información.
- e. Versiones de frameworks de desarrollo:
 - i. Angular JS 5.0.1
 - ii. Ruby 2.4.0
 - iii. Python 2.7.14
- 4. Las actividades ejecutadas durante los sprint se estimarán usando la técnica Planning Poker.
- 5. El ICFES entregará al contratista historias de usuario en la herramienta ICESCRUM, las cuales contendrán la siguiente información: Nombre breve y descriptivo, Descripción de la funcionalidad en forma de diálogo o monólogo del usuario describiendo la funcionalidad que desea realizar. Criterio de validación y verificación que determinará para considerar terminado y aceptable por el cliente el desarrollo de la funcionalidad descrita.

Y adicionalmente por la información que resulte necesaria por el modelo de implementación: Prioridad, Riesgo, Tamaño, entre otros. Los tiempos para aclaración de dudas serán acordados entre las partes.

ANEXOS

1. Anexo No. 1– Lista de Chequeo Arquitectura.
2. Anexo No. 2– Lineamientos Gráficos y de Usabilidad ICFES.

OSCAR YOVANY BAQUERO
Subdirector de Desarrollo de Aplicaciones