

# SICC

## Plan de verificación y validación

### Versión 1.3

#### *Historial de revisiones*

<b>FECHA</b>	<b>VERSIÓN</b>	<b>DESCRIPCIÓN</b>	<b>AUTOR</b>
<b>24/08/2016</b>	1.0	Creación del documento	Sergio Cabrera
<b>26/08/2016</b>	1.1	Actualización	Sergio Cabrera
<b>27/08/2016</b>	1.2	Cambios en las pruebas de performance	Sergio Cabrera
<b>28/08/2016</b>	1.3	Revision de SQA	Juan Nogueira
<b>08/09/2016</b>	2.0	Actualizacion	Sergio cabrera
<b>10/09/2016</b>	2.1	Cambio en el alcance, pruebas y cronograma	Sergio cabrera

# Contenido

<u>Introducción</u>	<u>3</u>
<u>Propósito</u>	<u>3</u>
<u>Punto de partida</u>	<u>3</u>
<u>Alcance</u>	<u>3</u>
<u>Identificación del proyecto</u>	<u>5</u>
<u>Estrategia de evolución del Plan</u>	<u>6</u>
<u>Requerimientos para verificar</u>	<u>6</u>
<u>Requerimientos funcionales</u>	<u>6</u>
<u>Requerimientos no funcionales</u>	<u>6</u>
<u>Estrategia de Verificación</u>	<u>6</u>
<u>Tipos de pruebas</u>	<u>7</u>
<u>Prueba de integridad de los datos y la base de datos</u>	<u>7</u>
<u>Prueba de Funcionalidad</u>	<u>7</u>
<u>Prueba de Ciclo del Negocio</u>	<u>8</u>
<u>Prueba de Interfase de Usuario</u>	<u>9</u>
<u>Prueba de Performance, Prueba de Carga, Prueba de Esfuerzo, Prueba de</u>	
<u>Volumen</u>	<u>10</u>
<u>Prueba de Seguridad y Control de Acceso</u>	<u>10</u>
<u>Prueba de Fallas y Recuperación</u>	
<u>11</u>	
<u>Prueba de Configuración</u>	<u>12</u>
<u>Prueba de Instalación</u>	<u>13</u>
<u>Prueba de Documentos</u>	<u>14</u>
<u>Herramientas</u>	<u>15</u>
<u>Recursos</u>	<u>15</u>
<u>Roles</u>	<u>15</u>
<u>Sistema</u>	<u>16</u>
<u>Hitos del proyecto de Verificación</u>	<u>17</u>
<u>Entregables</u>	<u>17</u>
<u>Modelo de Casos de Prueba</u>	<u>17</u>
<u>Informes de Verificación</u>	<u>17</u>
<u>Evaluación de la verificación</u>	<u>18</u>
<u>Informe final de verificación</u>	<u>18</u>
<u>Apéndice</u>	<u>18</u>
<u>Severidad y Prioridad</u>	<u>18</u>
<u>Criterio de ingreso al testing</u>	<u>19</u>
<u>Criterio de liberación por el equipo de testing</u>	<u>19</u>

# Introducción

## Propósito

Este Plan de Verificación para el proyecto SICC soporta los siguientes objetivos:

- Identificar la información de proyecto existente y los componentes de software que deben ser verificados.
- Enumerar los requerimientos recomendados para verificar.
- Recomendar y describir las estrategias de verificación que serán usadas.
- Identificar los recursos necesarios y proporcionar una estimación de esfuerzo para realizar la verificación.
- Enumerar los entregables del proyecto de verificación.

## Punto de partida

El objetivo de la verificación es detectar defectos y asegurar que los mismos sean corregidos antes de la liberación de la versión del producto de software. Además se debe comprobar que el mismo cumple con los requerimientos funcionales y no funcionales, así como también que la integración de los componentes que conforman el sistema, funcione adecuadamente.

La verificación se aplica a todos los componentes de desarrollo, así como también su integración e interacción.

El proyecto consiste en construir una aplicación móvil para el sistema operativo android, que consumirá servicios propuestos por la comisión de carrera. Estos servicios podrán incluir ver la escolaridad del estudiante, calcular las asignaturas que podrá cursar, contactarse con la comisión de carrera e información ampliada de las distintas asignaturas disponibles.

## Alcance

### *Pruebas de Componente*

Se realiza con el propósito de verificar la implementación de un elemento de software o una colección de los mismos. Permite asegurar que la lógica del programa es completa y correcta, y determina si el componente funciona tal cual fue diseñado.

La realización de las pruebas está a cargo de los desarrolladores. La herramienta escogida para la realización de pruebas de componente del servidor central es Unittest Módulo integrado en la biblioteca estándar de Python.

Debido a que el equipo de desarrollo está sometido a grandes cargas de trabajo y se sigue una metodología ágil (Scrum), no se pedirá al equipo de desarrollo que genere el informe de verificación unitaria.

### ***Pruebas de Integración:***

Compuestas por una progresión ordenada de pruebas en la cual los elementos de software/hardware son combinados y probados hasta lograr la integración completa del sistema. Permite asegurar que los objetivos del diseño son alcanzados, probando las interfaces de comunicación entre los distintos subsistemas.

La realización de las pruebas está a cargo del equipo de desarrollo. Para las mismas se aplicaran la técnica de integración por disponibilidad dado el perfil incremental propuesto por el cliente. Se creará una tabla de particiones válidas e inválidas para cada interfaz, contemplando en la misma los distintos parámetros de comunicación intercambiados.

Respecto a los criterios de aceptación, se espera que el conjunto de pruebas de integración cumpla con lo siguiente:

- Se verifica el comportamiento de todas las interfaces del sistema que impactan en los casos de uso críticos para la arquitectura.
- Las pruebas de humo seleccionadas por el equipo de verificación, conteniendo los requisitos de integración mínimos para satisfacer los flujos de funcionalidad positivos, se realizaron sin errores.

Debido a que el equipo de desarrollo está sometido a grandes cargas de trabajo y se sigue una metodología ágil (Scrum), se decidió que no se genere un informe de verificación de integración para estas pruebas.

### ***Pruebas funcionales***

Pretenden verificar que el sistema cumple los requisitos especificados. Asegura que el software como entidad completa cumple con sus requisitos operacionales.

Si el equipo de desarrollo cumple el criterio de ingreso a testing(ver), el equipo de verificación procederá a realizar las pruebas funcionales a partir de la ejecución de casos de prueba derivados a partir de los casos de uso.

Respecto a los criterios de aceptación, se espera que el conjunto de pruebas de sistema cumpla con lo siguiente:

- Verificación de los casos de uso por prioridad, la cual será establecida por el cliente según necesidad y por el arquitecto del proyecto según especificación de áreas críticas del sistema.
- Se ejecutaron todos los casos de prueba especificados para los casos de uso verificados.
- Todos los incidentes reportados en el proceso de verificación fueron corregidos y no se detectaron nuevos incidentes con el mismo conjunto de casos de prueba.

Debido a que es necesario evaluar la verificación, se pedirá al equipo de verificación que genere el informe de verificación del sistema para cada prueba que se realice.

## ***Pruebas no funcionales***

Se verifica que el sistema cumpla con los requerimientos no funcionales planteados por el cliente.

### **Pruebas de usabilidad**

Realizadas por el equipo de verificación, con el objetivo de determinar el nivel de calidad respecto a la usabilidad del sistema. Se tendrán en cuenta criterios de usabilidad establecidos por estándares de desarrollo de aplicaciones Android y web.

Se realizarán dos instancias de validación con usuarios finales, siendo la primera de ellas una instancia inicial del sistema mediante prototipos y la segunda con una versión más madura de la interfaz y el sistema en su conjunto.

Respecto a los criterios de aceptación, se tomará en cuenta que:

- Se respeten las pautas definidas y validadas por SQA en cuanto a usabilidad

### **Pruebas de seguridad**

Se controlará el acceso a la aplicación, tomando en cuenta los procesos de autenticación establecidos por el cliente. Por otro lado se controlará el criterio de encriptación de datos sensibles.

### **Pruebas de performance**

Se definirá en la próxima reunión con el cliente.

## ***Pruebas de Aceptación***

Prueba formal con el propósito de determinar si el sistema construido satisface los criterios de aceptación y permite al cliente determinar si el mismo será aceptado y puesto en su fase operacional. Asegura que los requisitos del cliente fueron cumplidos y que todos los componentes requeridos fueron incluidos en el sistema. Para lograr esto se realizará una reunión de validación con el cliente. En dicha reunión se acordará cuáles serán sus criterios de aceptación para cada requerimiento funcional y no funcional.

## **Identificación del proyecto**

Los documentos usados para elaborar el Plan de Verificación son los siguientes:

- Documento de especificación de requisitos.
- Plantilla de la memoria organizacional para el plan de verificación y validación.
- Memoria organizacional

## **Estrategia de evolución del Plan**

El encargado de monitorear el plan de verificación y validación es el responsable de verificación. Para lograr esto, dicha tarea tendrá el apoyo del equipo de verificación.

El plan deberá ser revisado periódicamente y se actualizará a medida que lo permitan los documentos de los cual depende.

Cualquier cambio que afecte el plan de verificación y validación deberá ser informado al responsable del mismo con el propósito de procesar y agendar dicha petición de cambio y así tomar las medidas necesarias.

En el caso de tratarse de un cambio menor, el mismo será aprobado directamente por el responsable de verificación. En el caso de tratarse de un cambio considerable se discutirá entre todos los verificadores.

La comunicación de los cambios al Plan de Verificación y Validación, se realizará mediante la forma habitual de comunicación (slack) de los integrantes del proyecto.

## **Requerimientos para verificar**

En la lista a continuación se presentan los requerimientos funcionales y requerimientos no funcionales, que serán verificados.

### **Requerimientos funcionales**

- Loguearse.
- Cerrar Sesión.
- Visualizar escolaridad.
- Visualizar materias habilitadas a cursar.
- Envío y configuración de alertas.

### **Requerimientos no funcionales**

- Seguridad de los datos.
- Facilidad de uso.
- Estética agradable.
- Tiempo de respuesta satisfactorio.

## **Estrategia de Verificación**

Esta sección presenta el enfoque recomendado para la verificación. Describe cómo se verificarán los elementos. Para cada tipo de prueba, se proporciona una descripción de la prueba y porque

será implementada y ejecutada. Si un tipo de prueba no será implementada y ejecutada, se indicará brevemente cual es la prueba que no se implementará o ejecutará y una justificación de ello. Se indicarán las técnicas usadas y el criterio para saber cuando una prueba se completó (criterio de aceptación). Las pruebas se deben ejecutar usando bases de datos conocidas y controladas en un ambiente seguro.

## **Tipos de pruebas**

### **Prueba de integridad de los datos y la base de datos**

#### *Justificación*

No se realizará este tipo de prueba debido a que la base de datos del cliente y la base de datos que usa el servidor intermedio(WSServer) solo admiten consultas de lectura por el momento. En caso que el cliente proponga el requisito de que un usuario pueda registrarse se tendrá en cuenta este tipo de prueba.

### **Prueba de Funcionalidad**

#### *Descripción*

La prueba de funcionalidad se enfoca en requerimientos para verificar que se corresponden directamente a casos de usos o funciones. Este tipo de prueba se basa en técnicas de caja negra, que consisten en verificar la aplicación y sus procesos interactuando con la aplicación por medio de la interfase de usuario y analizar los resultados obtenidos.

#### *Justificación*

Este tipo de prueba es necesario para garantizar que el software hace lo que debe y sobre todo, lo que se ha especificado en el documento de requisitos.

#### *Objetivo de la prueba*

Asegurar la funcionalidad apropiada del objeto de prueba, incluyendo la navegación, entrada de datos, proceso y recuperación.

#### *Técnica*

Se usarán los casos de pruebas diseñados a partir de los casos de uso, luego se ingresan los datos de pruebas(válidos e inválidos) en la interfaz del dispositivo mientras se ejecuta la herramienta Appium o Selenium, el cual mediante el inspector se encargará de crear código para cada acción del tester. Si el software no hace lo que debe entonces se debe reportar el error al equipo de desarrollo y guardar el código generado para automatizar la prueba.

#### *Criterio de aceptación*

Todas las pruebas planificadas se realizaron. Todos los defectos encontrados han sido debidamente identificados.

#### *Consideraciones especiales*

Identificar o describir aquellos elementos o problemas (internos o externos) que impactaron en la implementación y ejecución de las pruebas de funcionalidad.

## Prueba de Ciclo del Negocio

### *Descripción*

Esta prueba debe simular las actividades realizadas en el proyecto en el tiempo. Se debe identificar un período, que puede ser un año, y se deben ejecutar las transacciones y actividades que ocurrirían en el período de un año. Esto incluye todos los ciclos diarios, semanales y mensuales y eventos que son sensibles a la fecha.

### *Justificación*

Este tipo de prueba no se realizará, porque el sistema software a construir no tiene un ciclo de negocio definido en el documento de requisitos. Un ejemplo donde se podría ejecutar este tipo de prueba es en un sistema de administración (o contable), donde se genera una plantilla de pago cada año

## Prueba de Interfase de Usuario

### *Descripción*

Esta prueba verifica que la interfase de usuario proporcione al usuario el acceso y navegación a través de las funciones apropiada. Además asegura que los objetos presentes en la interfase de usuario se muestran como se espera y conforme a los estándares establecidos por la empresa o de la industria.

### *Justificación*

Se realizará este tipo de prueba debido a que la aplicación debe cumplir el requerimiento no funcional de usabilidad.

### *Objetivo de la prueba*

Verificar que: la aplicación sea de fácil uso y que la navegación a través de los elementos que se están probando reflejen los requerimientos, incluyendo manejo de ventanas, campos y métodos de acceso; los objetos de las ventanas y características, como menús, tamaño, posición, estado funcionen de acuerdo a los estándares mínimos definidos.

### *Técnica*

Para verificar la usabilidad se usará la técnica thinking aloud (pensando en voz alta). La técnica consiste en proporcionar a los usuarios el prototipo a probar y un conjunto de tareas a realizar.

- Se les pide que realicen las tareas y que expliquen en voz alta qué es lo que piensan al respecto mientras están trabajando con la interfaz, describiendo qué es lo que creen que está pasando, por qué toman una u otra acción o qué es lo que están intentando realizar. En definitiva, ¡cuantas más impresiones mejor!
- Pensando en voz alta permite a los evaluadores comprender cómo el usuario se aproxima al objetivo con la interfaz propuesta y qué consideraciones tiene en la mente cuando la usa. El usuario puede expresar que la secuencia de etapas que le dicta el producto para realizar el objetivo de su tarea es diferente de la que esperaba.

Para verificar que se cumple el estándar, crear o modificar pruebas para cada ventana verificando la navegación y los estados de los objetos para cada ventana de la aplicación y cada objeto dentro de la ventana.

#### *Criterio de aceptación*

La aplicación cumple con las métricas de usabilidad definida en el plan de calidad. Cada ventana ha sido verificada exitosamente siendo consistente con una versión de referencia o estándar establecido.

#### *Consideraciones especiales*

Tener en cuenta la experiencia de usuario en cuanto a amigabilidad, intuitividad, baja curva de aprendizaje y facilidad de uso.

### **Prueba de Performance, Prueba de Carga, Prueba de Esfuerzo, Prueba de Volumen**

#### *justificación*

No se realizarán este tipo de pruebas, debido a que la aplicación a construir no es un sistema que esté pensado (por la naturaleza del sistema) para manejar grandes volúmenes de tráfico.

### **Prueba de Seguridad y Control de Acceso**

#### *Descripción*

La Prueba de Seguridad y Control de Acceso se enfoca en dos áreas de seguridad:

- Seguridad en el ámbito de aplicación, incluyendo el acceso a los datos y a las funciones de negocios.
- Seguridad en el ámbito de sistema, incluyendo conexión, o acceso remoto al sistema.

La seguridad en el ámbito de aplicación asegura que, basado en la seguridad deseada los actores están restringidos a funciones o casos de uso específicos o limitados en los datos que están disponibles para ellos.

La seguridad en el ámbito de sistema asegura que, solo los usuarios con derecho a acceder al sistema son capaces de acceder a las aplicaciones y solo a través de los puntos de ingresos apropiados.

#### *Justificación*

Se realizará este tipo de prueba debido a que la aplicación debe cumplir el requerimiento no funcional de seguridad.

#### *Objetivo de la prueba*

Seguridad en el ámbito de aplicación: Verificar que un actor pueda acceder solo a las funciones o datos para los cuales su tipo de usuario tiene permiso.

Seguridad en el ámbito de sistema: Verificar que solo los actores con acceso al sistema y a las aplicaciones, puedan acceder a ellos.

Encriptación de datos sensibles:

- Verificar métodos de encriptación y validar que los datos son enviados entre los componentes del sistema con el encriptado correcto.

#### *Técnica*

Seguridad en el ámbito de aplicación: Identificar y hacer una lista de cada tipo de usuario y las funciones y datos sobre las que cada tipo tiene permiso.

Crear pruebas para cada tipo de usuario y verificar cada permiso creando operaciones específicas para cada tipo de usuario.

Modificar el tipo de usuario y volver a ejecutar las pruebas para los mismos usuarios. En cada caso, verificar que las funciones o datos adicionales están correctamente disponibles o son denegados.

Verificar métodos de encriptación y validar que los datos son enviados con el encriptado correcto

Acceso en el ámbito de sistema: Ver consideraciones especiales más abajo.

#### *Criterio de aceptación*

Para cada tipo de actor conocido las funciones y datos apropiados están disponibles, y todas las operaciones funcionan como se espera y ejecutan las pruebas de Funcionalidad de la aplicación.

#### *Consideraciones especiales*

El acceso al sistema debe ser discutido con el administrador del sistema o la red. Esta prueba no puede requerirse como tal, es una función del administrador del sistema o de la red.

## **Prueba de Fallas y Recuperación**

### *Descripción*

La Prueba de Recuperación es un proceso en el cual la aplicación o sistema se expone a condiciones extremas, o condiciones simuladas, para causar falla, como fallas en dispositivos de Entrada/Salida o punteros a la base de datos inválidos. Los procedimientos de recuperación se invocan y la aplicación o sistema es monitoreado e inspeccionado para verificar que se recupera apropiadamente la aplicación o sistema y se logre la recuperación de datos.

### *Justificación*

A pesar de que no es un requisito no funcional se tendrá en cuenta para asegurar que el software puede recuperarse de fallas de hardware, software o mal funcionamiento de la red sin pérdida de datos o de integridad de los datos.

### *Objetivo de la prueba*

Verificar que los procesos de recuperación (manual o automáticos) recuperen apropiadamente la base de datos, aplicaciones y sistema a un estado conocido y deseado. En la prueba se incluyen los siguientes tipos de condiciones:

- interrupción de energía al cliente
- interrupción de energía al servidor

- interrupción de comunicaciones mediante los servidores de la red
- interrupción de comunicación o pérdida de energía de los discos del servidor o con los controladores
- ciclos incompletos (procesos de filtrado de datos interrumpidos, procesos de sincronización de datos interrumpidos)
- punteros a la base de datos o claves inválidos
- elementos de datos en la base de datos inválidos o corruptos.

### *Técnica*

Se deben usar las pruebas creadas para probar Funcionalidad para crear una serie de operaciones. Una vez logrado el punto de comienzo deseado, se deben realizar o simular las siguientes acciones, individualmente:

- Interrumpir la energía del cliente: apagar el PC.
- Interrumpir la energía del servidor: simular o iniciar el proceso de apagado del servidor.
- Interrupción por medio de los servidores de red: simular o iniciar la pérdida de comunicación con la red (desconectar físicamente la comunicación o apagar el servidor de red o router)
- Interrumpir la comunicación o quitar la energía de los discos del servidor o sus controladores: simular o eliminar físicamente al comunicación con uno o más controladores de disco o los discos.
- Una vez que se lograron o simularon estas condiciones, se deben invocar los procedimientos de recuperación.
- Las pruebas de ciclos incompletos utilizan la misma técnica excepto que los procesos de bases de datos deben ser abortados a sí mismos o terminados prematuramente.
- Las últimas dos pruebas requieren que se logre un estado conocido de la base de datos. Se deben corromper manualmente campos de la base de datos, punteros y claves trabajando directamente sobre la base de datos (utilizando herramientas para la base de datos). Se deben ejecutar las pruebas de Funcionalidad y Ciclo de negocio y verificar que los ciclos se completen.

### *Criterio de aceptación*

En todos los casos, la aplicación, la base de datos y el sistema deben, en la realización procedimientos de recuperación, volver a un estado conocido y deseable. Este estado incluye corrupción de datos limitada al los campos, punteros o claves corruptos conocidos, y reportes indicando los procesos u operaciones que no se completaron debido a las interrupciones.

### *Consideraciones especiales*

Los procedimientos para desconectar cables (simulando falta de energía o pérdida de comunicación) no son deseables o factibles. Se pueden requerir métodos alternativos, como software de diagnóstico. Se requieren los grupos de recursos de Sistemas, Bases de datos y Red.

Estas pruebas deben ejecutarse fuera del horario de trabajo normal o en una máquina aislada.

## Prueba de Configuración

La Prueba de Configuración verifica el funcionamiento del software con diferentes configuraciones software y hardware.

### *Justificación*

Se realizará este tipo de prueba, debido a que la aplicación hecha en android nativo debe funcionar en distintos dispositivos que usen la misma versión del sistema operativo android.

### *Objetivo de la prueba*

Verificar que el software funcione apropiadamente en las configuraciones requeridas de hardware y software.

### *Técnica*

Usar las pruebas de Funcionalidad.

- Abrir y cerrar varias sesiones de software que no son objeto de prueba, como parte de la prueba o antes de comenzar la prueba.
- Ejecutar operaciones seleccionadas para simular la interacción del actor con el software objeto de prueba y con el software que no es objeto de prueba.
- Repetir los procedimientos anteriores minimizando la memoria convencional disponible en la máquina cliente.

### *Criterio de aceptación*

Por cada combinación de software objeto de prueba y software que no es objeto de prueba, todas las operaciones son completadas exitosamente sin fallas.

### *Consideraciones especiales*

Todo el software que no es objeto de prueba que es necesario y debe estar accesible.

¿Qué aplicaciones se usan normalmente?

¿Qué información se maneja en las aplicaciones que se usan normalmente, y que tamaño de información?.

## Prueba de Instalación

La Prueba de Instalación tiene dos propósitos. Uno es asegurar que el software puede ser instalado en diferentes condiciones (como una nueva instalación, una actualización, y una instalación completa o personalizada) bajo condiciones normales y anormales. Condiciones anormales pueden ser insuficiente espacio en disco, falta de privilegios para crear directorios, etc. El otro propósito es verificar que, una vez instalado, el software opera correctamente. Esto significa normalmente ejecutar un conjunto de pruebas que fueron desarrolladas para Prueba de Funcionalidad.

### *Justificación*

Evitar que surjan errores al momento de instalar la aplicación.

#### *Objetivo de la prueba*

Verificar que el software objeto de prueba se instala correctamente en cada configuración de hardware requerida bajo las siguientes condiciones:

- instalación nueva, una nueva máquina, nunca instalada previamente con SICC
- actualización, máquina previamente instalada con SICC, con la misma versión
- actualización, máquina previamente instalada con SICC, con una versión anterior.

#### *Técnica*

Manualmente o desarrollando programas, para validar la condición de la máquina destino (nueva, nunca instalado, misma versión, versión anterior ya instalada). Realizar la instalación, por último ejecutar un conjunto de pruebas funcionales ya implementadas para la Prueba de Funcionalidad.

#### *Criterio de aceptación*

Las pruebas de funcionalidad de SICC se ejecutan exitosamente sin fallas.

#### *Consideraciones especiales*

N/A.

### **Prueba de Documentos**

#### *Justificación*

La Prueba de Documentos es necesario para evitar inconsistencias, por lo que se debe asegurar que los documentos relacionados al software que se generen en el proceso sean correctos, consistentes y entendibles.

#### *Objetivo de la prueba*

Verificar que el documento objeto de prueba sea:

- Correcto, esto es, que cumpla con el formato y organización para el documento establecido en el proyecto.
- Consistente, esto es, que el contenido del documento sea fiel a lo que hace referencia. Si el documento es Documentación de Usuario, que la explicación de un procedimiento sea exactamente como se realiza el procedimiento en el software, si se muestran pantallas que sean las correctas.
- Entendible, esto es, que al leer el documento se entienda correctamente lo que expresa y sin ambigüedades, además que sea fácil de leer.

#### *Técnica*

Para verificar que el documento es correcto se debe comparar con el estándar definido si existe o con las pautas de documentación y ver que el documento cumple con ellas.

Para verificar que el documento es Consistente se debe ejecutar el programa siguiendo el documento en caso de los Materiales de Soporte al Usuario y comprobar que lo que se explica en estos documentos es exactamente lo que se ejecuta en el programa. En caso de Documentación Técnica se debe revisar el código al cual corresponde la documentación y comprobar que dicha describe el código.

Para verificar que el documento es entendible, debe comprobar que se entiende correctamente, que no tiene ambigüedades y que sea fácil de leer.

### *Criterio de aceptación*

El documento expresa exactamente lo que debe expresar, no hay diferencias entre lo que está escrito y el objeto de la descripción (operación de software, código de programa, decisiones técnicas) y se entiende fácilmente.

### *Consideraciones especiales*

N/A.

## **Herramientas**

Postgresql

Servirá para establecer la base de datos en caso de que esta esté corrupta.

unitTest

herramienta para las pruebas unitarias y de integración.

python3-coverage

Permite visualizar la cobertura de las pruebas unitarios.

Jira:

Software para el seguimiento de incidentes.

Selenium:

Herramienta para automatizar las pruebas funcionales en la aplicación web.

Appium:

Herramienta para automatizar las pruebas funcionales de la aplicación android.

IDE :

Android Studio.

## **Recursos**

En esta sección se presentan los recursos recomendados para el proyecto SICC, sus principales responsabilidades y su conocimiento o habilidades.

## Roles

En la tabla a continuación se muestra la composición de personal para el proyecto SICC en el área Verificación del Software.

<b>Rol</b>	<b>Cantidad mínima de recursos recomendada</b>	<b>Responsabilidades</b>
Responsable de verificación	1	Identifica, prioriza y diseña los casos de prueba. Genera el Plan de Verificación. Genera el Modelo de Prueba. Evalúa el esfuerzo necesario para verificar. Proporciona la dirección técnica. Adquiere los recursos apropiados. Proporciona informes sobre la verificación.
Asistente de verificación	3	Diseña casos de prueba Ejecuta las pruebas Registra los resultados de las pruebas. Documenta los pedidos de cambio.
Administrador de Base de Datos	1	Realiza la gestión y mantenimiento del entorno de los datos (base de datos) de prueba y los recursos. Administra la base de datos de prueba.

## Sistema

En la siguiente tabla se establecen los recursos de sistema necesarios para realizar la verificación.

Es recomendable que el sistema simule el entorno de producción, reduciendo los accesos y los tamaños de bases de datos si fuera apropiado.

Tanto para aplicación android y web se utilizarán los siguientes servidores:

- WSServer: Servidor intermedio.
- Central server: Servidor proporcionado por el cliente.

<b>Aplicacion Android nativo</b>	
<b>Recurso</b>	<b>Nombre/Tipo</b>
Emulador Android	Android Studio
Dispositivos mobile para pruebas	Se definirá con el cliente en la próxima reunión.
Versión del sistema operativo de android en donde se ejecutarán las pruebas.	Se definirá con el cliente en la próxima reunión.
Aplicación móvil	SICC
<b>Aplicacion web Responsive</b>	
<b>Recurso</b>	<b>Nombre/Tipo</b>
Browser	Firefox, Chrome (ambas en su última versión)
Aplicacion web	SICC (WebClient)
Dispositivos mobile para pruebas	Se definirá con el cliente en la próxima reunión.

## Hitos del proyecto de Verificación

Los hitos del Proyecto para el Sprint 2 son los siguientes

Hito	Esfuerzo en horas	inicio - fin
Plan de verificación de la Iteración	3 hs	13/09/2016 - 13/09/2016
Plan de verificación y validación	2 hs	14/09/2016 - 14/09/2016
Modelo de casos de prueba	6 hs	14/09/2016 - 16/09/2016
Informe de verificación del sistema	10 hs	19/09/2016 - 22/09/2016

Evaluación de la verificación del sistema	8 hs	22/09/2016 - 24/09/2016
-------------------------------------------	------	-------------------------

## Entregables

En esta sección se lista los documentos, herramientas e informes que se crearán, por quién, para quién. Para cada entregable deberá indicar las fechas en que son liberadas todas las versiones del mismo.

### Modelo de Casos de Prueba

Documento	Modelo de Casos de Prueba
Creado por	El Responsable de verificación, Sergio Cabrera
Para quién	Es la guía para realizar las pruebas del sistema y lo usarán los Asistentes de verificación y el Responsable de verificación cuando se ejecuten las pruebas del sistema.
Fecha de liberación	16/09/2016

### Informes de Verificación de sistema

Documento	Se genera un documento Informe de Verificación de Sistema por cada prueba de sistema que se realice.
Creado por	Equipo de testing.
Para quién	Es el retorno para los implementadores de la tarea de verificación, que detalla los errores encontrados para que puedan ser corregidos.
Fecha de liberación	Será liberado luego de cada verificación de sistema.

### Evaluación de la verificación

Documento	Se genera un documento Evaluación de la verificación por cada prueba que se realice al sistema. Este documento contiene las fallas encontradas en el sistema, la cobertura de la verificación realizada y el estado del sistema.
-----------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Creado por	El Responsable de verificación, que toma como fuente de su trabajo el informe de verificación del sistema.
Para quién	Es el resumen de la tarea de verificación y es el retorno para todo el equipo de trabajo del estado del sistema.
Fecha de liberación	24/09/2016

## Informe final de verificación

Documento	El documento Informe final de verificación es el resumen de la verificación final del sistema antes de que sea liberado al entorno del usuario.
Creado por	El Responsable de verificación, que toma como fuente de su trabajo los Informes de verificación.
Para quién	Indica el estado del sistema.
Fecha de liberación	Será liberado luego de la verificación final del sistema.

## Apéndice

### Severidad y Prioridad

severidad	Descripcion
<b>1</b>	<b>Fatal / Crash / Bloquer / Crítico:</b> El sistema colapsa. No es posible continuar testeando la aplicación.
<b>2</b>	<b>Mayor:</b> Se afectan funcionalidades vitales para el sistema. De gran impacto para el negocio.
<b>3</b>	<b>Menor:</b> Se afectan funcionalidades menores del sistema que no significan gran impacto para el negocio.
<b>4</b>	<b>Trivial / mejoras / Sugerencias:</b> Sugerencias de apariencia, usabilidad, etc. El sistema funciona correctamente.

### Criterio de ingreso al testing

Antes de que los productos de software sean entregados al equipo de testing, se espera que:

- Los productos de software ya han sido completamente desarrollados.
- El testing unitario y de integración a resultado exitoso.
- El ambiente de pruebas (incluidas las herramientas de testing) ya ha sido correctamente instalado y configurado en todas las máquinas a utilizar.

### **Criterio de liberación por el equipo de testing**

El criterio general para que testing considere que el producto se encuentre en condiciones de que el producto sea entregado al cliente, debe cumplir lo siguiente:

- Todas las tareas de testing planificadas han sido completadas bajo los lineamientos acordados.
- Los defectos de severidad 1 y 2 han sido corregidos, re-testeados y cerrados.