

SICC

Plan de verificación y validación

Versión 3.2

Historial de revisiones

FECHA	VERSIÓN	DESCRIPCIÓN	AUTOR
24/08/2016	1.0	Creación del documento	Sergio Cabrera
26/08/2016	1.1	Actualización	Sergio Cabrera
27/08/2016	1.2	Cambios en las pruebas de performance	Sergio Cabrera
28/08/2016	1.3	Revision de SQA	Juan Nogueira
08/09/2016	2.0	Actualización	Sergio Cabrera
10/09/2016	2.1	Cambio en el alcance, pruebas y cronograma	Sergio Cabrera
11/09/2016	2.1	Revision de SQA	Juan Nogueira
25/09/2016	3.0	Actualización de recursos	Sergio Cabrera
26/09/2016	3.1	Revisión del documento	Belén Techera
08/10/2016	3.2	Actualización de alcance, herramientas de automatización, requerimientos funcionales, pruebas funcionales y prueba de seguridad.	Sergio Cabrera

Contenido

Introducción

Propósito

Punto de partida

Alcance

Identificación del proyecto

Estrategia de evolución del Plan

Requerimientos para verificar

Requerimientos funcionales

Requerimientos no funcionales

Estrategia de Verificación

Tipos de pruebas

Prueba de integridad de los datos y la base de datos

Prueba de Funcionalidad

Prueba de Ciclo del Negocio

Prueba de Interfase de Usuario

Prueba de Performance, Prueba de Carga, Prueba de Esfuerzo, Prueba de Volumen

Prueba de Seguridad y Control de Acceso

Prueba de Fallas y Recuperación

Prueba de Configuración

Prueba de Instalación

Prueba de Documentos

Herramientas

Recursos

Roles

Sistema

Entregables

Modelo de Casos de Prueba

Informes de Verificación

Evaluación de la verificación

Informe final de verificación

Apéndice

Severidad y Prioridad

Criterio de ingreso al testing

Criterio de liberación por el equipo de testing

Introducción

Propósito

El 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 brindados 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 (híbrido entre Scrum y MUM), no se pedirá al mismo que genere un 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 aplicarán la técnica de integración por disponibilidad debido al 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, 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 al testing, el equipo de verificación procederá a ejecutar los casos de prueba derivados a partir de los casos de uso. Para el desarrollo de los casos de prueba se utilizarán técnicas de caja negra y para su implementación se usarán las [herramientas](#) de automatización como codeception y selenium server. Además se ejecutarán pruebas de regresión para evitar que los cambios introducidos al software no alteren las funcionalidades que ya habían pasado la prueba.

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 y se evitará que el sistema sufra de ataques sql injection .

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

1. Login: se creará una base de datos de pruebas que contenga usuario y contraseña y se validará el acceso de los usuarios contra la misma.

2. Cerrar Sesión: con esta funcionalidad el usuario quedará deslogueado.
3. Visualizar escolaridad: esta funcionalidad, como su nombre lo indica, permite que cada estudiante pueda ver su escolaridad.
4. Visualizar exámenes pendientes: esta funcionalidad permite que cada estudiante pueda ver los exámenes que tiene pendientes.
5. Visualizar materias habilitadas a cursar: esta funcionalidad permite que un estudiante pueda ver qué materias puede cursar según con las materias que tenga aprobadas.
6. Planificador de carrera: Esta funcionalidad permite planificar qué asignaturas desea cursar semestre a semestre el estudiante.
7. Envío y configuración de notificaciones: el sistema permite enviar alertas a los usuarios acerca de vencimientos de cursos y exámenes pendientes cuando se acerque un período de examen.
8. Un estudiante puede estar inscrito a más de una carrera y debería poder ver su escolaridad y su planificación para cada una de ellas. Para lo mismo deberá cambiar la configuración de la aplicación desde la ventana correspondiente.

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, y se realizará la implementación de cada caso de prueba con la herramienta codeception, luego se ejecutarán los casos de prueba. Si el software no pasa las pruebas 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 en el ámbito de seguridad en la aplicación, debido a que la aplicación debe cumplir el requerimiento no funcional de seguridad y privacidad. Sin embargo la seguridad en el ámbito de sistema no se realizará (Ver consideraciones especiales más abajo).

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 tiene permiso y verificar que los datos del usuario son encriptados.

Técnica

Para verificar que los datos viajan encriptados se utilizará el analizador de tráfico wireshark. Además para descubrir que la aplicación tiene vulnerabilidad sql injection se ejecutarán pruebas con la herramienta zap (zed attack proxy) e inspecciones al código para comprobar si existen consultas sql mal programadas.

Criterio de aceptación

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

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

N/A

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.

Karma

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.

IDE :

Android Studio.

Codeception :

Framework que nos permite codificar los casos de prueba en el lenguaje de php.

Selenium Server :

Es un servidor que maneja los drivers del navegador y permite aceptar peticiones remotas para manejar dicho navegador.

Wireshark :

Analizador de trafico de paquetes.

Zap :

Herramienta que sirve para realizar pruebas de seguridad(sql injection).

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.

Rol	Cantidad mínima de recursos recomendada	Responsabilidades
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.

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.
- Base de datos sicc para la autenticación.
- Central server: Servidor proporcionado por el cliente.

Aplicacion Android nativo	
Recurso	Nombre/Tipo
Emulador Android	Android Studio
Versión del sistema operativo de android en donde se ejecutarán las pruebas.	Superior o igual a la versión 4.0 de android
Aplicación móvil	SICC
Aplicacion web Responsive	
Recurso	Nombre/Tipo
Browser	Firefox, Chrome, Safari (todas en su última versión)
Aplicacion web	SICC (WebClient)
Dispositivos para pruebas	tablets, móvil y pc .

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	Aun no definida

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.

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
1	Fatal / Crash / Bloquer / Critico: El sistema colapsa. No es posible continuar testeando la aplicación.
2	Mayor: Se afectan funcionalidades vitales para el sistema. De gran impacto para el negocio.
3	Menor: Se afectan funcionalidades menores del sistema que no significan gran impacto para el negocio.
4	Trivial / mejoras / Sugerencias: 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.