

Challenge Accepted

Plan de Verificación y Validación

Versión 2.1

Historia de revisiones

Fecha	Versión	Descripción	Autor
28/08/2014	1.0	Primera versión del Documento.	Gonzalo Sintas.
29/08/2014	1.1	Actualización del Documento.	Gonzalo Sintas.
30/08/2014	1.2	Actualización del Documento.	Gonzalo Sintas.
31/08/2014	1.3	Revision final de SQA.	Federico Blumetto.
13/09/2014	2.0	Actualización del Plan.	Gonzalo Sintas
14/09/2014	2.1	Revision de SQA.	Federico Blumetto.

Contenido:

1. Introducción
 - 1.1. Propósito
 - 1.2. Punto de partida
 - 1.3. Alcance
 - 1.4. Identificación del proyecto
 - 1.5. Estrategia de evolución del Plan
2. Requerimientos para verificar
3. Estrategia de Verificación
 - 3.1. Tipos de pruebas
 - 3.1.1. Prueba de integridad de los datos y la base de datos
 - 3.1.1.1. Objetivo de la prueba
 - 3.1.1.2. Técnica
 - 3.1.1.3. Criterio de aceptación
 - 3.1.1.4. Consideraciones especiales
 - 3.1.2. Prueba de Funcionalidad
 - 3.1.2.1. Objetivo de la prueba
 - 3.1.2.2. Técnica
 - 3.1.2.3. Criterio de aceptación
 - 3.1.2.4. Consideraciones especiales
 - 3.1.3. Prueba de Ciclo del Negocio
 - 3.1.3.1. Objetivo de la prueba
 - 3.1.3.2. Técnica
 - 3.1.3.3. Criterio de aceptación
 - 3.1.3.4. Consideraciones especiales
 - 3.1.4. Prueba de Interfase de Usuario
 - 3.1.4.1. Objetivo de la prueba
 - 3.1.4.2. Técnica
 - 3.1.4.3. Criterio de aceptación
 - 3.1.4.4. Consideraciones especiales
 - 3.1.5. Prueba de Performance
 - 3.1.5.1. Objetivo de la prueba
 - 3.1.5.2. Técnica
 - 3.1.5.3. Criterio de aceptación
 - 3.1.5.4. Consideraciones especiales
 - 3.1.6. Prueba de Carga
 - 3.1.6.1. Objetivo de la prueba
 - 3.1.6.2. Técnica
 - 3.1.6.3. Criterio de aceptación
 - 3.1.6.4. Consideraciones especiales
 - 3.1.7. Prueba de Esfuerzo (stress, competencia por recursos, bajos recursos)
 - 3.1.7.1. Objetivo de la prueba

- [3.1.7.2. Técnica](#)
 - [3.1.7.3. Criterio de aceptación](#)
 - [3.1.7.4. Consideraciones especiales](#)
 - [3.1.8. Prueba de Volumen](#)
 - [3.1.8.1. Objetivo de la prueba](#)
 - [3.1.8.2. Técnica](#)
 - [3.1.8.3. Criterio de aceptación](#)
 - [3.1.8.4. Consideraciones especiales](#)
 - [3.1.9. Prueba de Seguridad y Control de Acceso](#)
 - [3.1.9.1. Objetivo de la prueba](#)
 - [3.1.9.2. Técnica](#)
 - [3.1.9.3. Criterio de aceptación](#)
 - [3.1.9.4. Consideraciones especiales](#)
 - [3.1.10. Prueba de Fallas y Recuperación](#)
 - [3.1.10.1. Objetivo de la prueba](#)
 - [3.1.10.2. Técnica](#)
 - [3.1.10.3. Criterio de aceptación](#)
 - [3.1.10.4. Consideraciones especiales](#)
 - [3.1.11. Prueba de Configuración](#)
 - [3.1.11.1. Objetivo de la prueba](#)
 - [3.1.11.2. Técnica](#)
 - [3.1.11.3. Criterio de aceptación](#)
 - [3.1.11.4. Consideraciones especiales](#)
 - [3.1.12. Prueba de Instalación](#)
 - [3.1.12.1. Objetivo de la prueba](#)
 - [3.1.12.2. Técnica](#)
 - [3.1.12.3. Criterio de aceptación](#)
 - [3.1.12.4. Consideraciones especiales](#)
 - [3.1.13. Prueba de Documentos](#)
 - [3.1.13.1. Objetivo de la prueba](#)
 - [3.1.13.2. Técnica](#)
 - [3.1.13.3. Criterio de aceptación](#)
 - [3.1.13.4. Consideraciones especiales](#)
 - [3.2. Herramientas](#)
- [4. Recursos](#)
 - [4.1. Roles](#)
 - [4.2. Sistema](#)
- [5. Hitos del proyecto de Verificación](#)
- [6. Entregables](#)
 - [6.1. Modelo de Casos de Prueba](#)
 - [6.2. Informes de Verificación](#)
 - [6.3. Evaluación de la verificación](#)
 - [6.4. Informe final de verificación](#)

7. Dependencias

7.1. Dependencia de personal

7.2. Dependencia de software

7.3. Dependencia de hardware

7.4. Dependencia de datos y base de datos de prueba

8. Riesgos

8.1. Planificación

8.2. Técnico

8.3. Gestión

9. Apéndice

9.1. Niveles de gravedad de error

9.2. Niveles de aceptación para los elementos verificados

1. Introducción

1.1. Propósito

Este Plan de Verificación para el proyecto "Challenge Accepted" soporta los siguientes objetivos:

- Identificar y analizar la información existente del proyecto.
- Decidir cuáles serán los componentes de software del proyecto sobre los que se realizarán pruebas de verificación y validación pertinentes.
- Listar aquellos requerimientos documentados en la especificación de requerimientos que deberán ser validados en el proceso, así como marcar la prioridad para cada uno.
- Describir cuáles serán las estrategias de verificación y validación a seguir y cuál es el objetivo que persigue cada una de ellas.
- Presentar los recursos técnicos y humanos que serán necesarios para llevar adelante el Plan de Verificación y Validación.
- Estimar el esfuerzo necesario para desarrollar las pruebas detalladas en este plan, para ello, el grupo de verificación estará compuesto por:
 - Gonzalo Sintas Responsable de Verificación.
 - Federico BLumetto Asistente de Verificación.
 - Inés Saint Martin Asistente de Verificación.
 - Lucas Bouissa Asistente de Verificación.
 - Florencia Ruiz Asistente de Verificación.

Se deberá tener a consideración que los miembros del grupo también cumplan tareas en otras áreas.

- Definir la documentación que se generará, su función dentro del plan y que representa cada una.

1.2. Punto de partida

Este Plan tiene como foco central la verificación y validación de las aplicaciones móviles que se desarrollarán para cumplir con el pedido del cliente de crear un juego de desafíos multijugador. Además de las aplicaciones, será de interés realizar pruebas sobre el servidor y la base de datos del sistema, así como la comunicación entre los distintos componentes, para verificar y validar su correcto funcionamiento.

El objetivo principal de este plan es poder demostrarle fehacientemente al cliente que el software desarrollado cumple con los requerimientos estipulados previamente.

1.3. Alcance

Dentro del alcance del Plan de Verificación y Validación se realizarán las siguientes tareas:

- Pruebas de Desarrollo: las mismas estarán a cargo del grupo de desarrollo y su principal objetivo será la detección de defectos (bugs) en el software de forma temprana. Las podemos categorizar en dos tipos:
 - Pruebas Unitarias: Se realizarán Pruebas Unitarias para verificar el correcto funcionamiento de los elementos desarrollados.
 - Pruebas de Componentes: Enfocada en probar el correcto funcionamiento de cada uno de los componentes del sistema, así como sus interfaces.
- Pruebas de Versión: A diferencia de las pruebas de desarrollo, las pruebas de versión estarán a cargo de un grupo de pruebas independiente. El objetivo de estas pruebas es demostrar que el software desarrollado cumple con las especificaciones de requerimientos acordadas con el cliente. Apreciamos dos tipos de Pruebas de Versión:
 - Pruebas de Requerimientos: enfocadas en validar los requerimientos acordados.
 - Pruebas de Escenario: se desarrollarán escenarios realistas de ejecución para validar que el comportamiento integrado sea el correcto.
- Pruebas de Usuario: en caso de existir disponibilidad, se estudiará la posibilidad de realizar, una vez la versión final supere el proceso de pruebas de desarrollo y versión, Pruebas de Usuario para obtener retroalimentación del software desarrollado.

Para poder cumplir con estas pruebas será necesario explorar diversas estrategias de verificación.

Para las Pruebas de Desarrollo, al dividirse la implementación en el desarrollo de casos de uso, se utilizará como criterio para las pruebas unitarias la prueba del funcionamiento de todos los flujos del caso de uso implementado. A su vez, las pruebas de Componentes, se enfocarán, una vez generada la integración del componente, en utilizar pruebas de humo para garantizar que la integración de los diversos subsistemas sea la correcta.

En las Pruebas de Versión el grupo de verificación realizará casos de prueba de diversos tipos para probar que se cumpla con los requerimientos. A su vez, se

buscará poder realizar pruebas Automatizadas al sistema para pruebas de regresión, pruebas de errores conocidos (para ver si fueron solucionados) y, de ser posible, utilizarlas para generar pruebas de escenarios.

El Plan de Verificación y Validación no tendrá como objetivo validar aspectos estéticos del software desarrollado, ni requerimientos no funcionales enfocados en la fluidez de las aplicaciones.

Este plan se actualizará conforme el diseño de la aplicación vaya “tomando forma”, siendo que cambios en los requerimientos (nuevos o requerimientos que se dan de baja), problemas de incompatibilidad o licenciamiento de las tecnologías a aplicar o modificación de los recursos puede alterar el mismo.

1.4. Identificación del proyecto

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

- Template del Plan de Verificación y Validación que se encuentra en el Modelo de Proceso MUM.
- Documento de Especificación de Requerimientos.
- Actas de las reuniones con el Cliente.
- Distintos Planes de Verificación y Validación de años anteriores tomados de la Memoria Organizacional.
- Modelos de Casos de Uso - Aún no documentado.
- Documento de Riesgos - Aún no documentado.

1.5. Estrategia de evolución del Plan

El Plan de Verificación y Validación será responsabilidad del Responsable de Verificación (Gonzalo Sintas) y su revisión será bisemanal, salvo frente a la ocurrencia de hechos excepcionales que ameriten realizar una revisión en un menor plazo.

Las revisiones podrán llevar a realizar modificaciones al Plan. Los mismos deberán ser evaluados y aprobados por el administrador y serán comunicados inmediatamente al grupo de Verificación via mail.

2. Requerimientos para verificar

En la lista a continuación se presentan los elementos, casos de uso, requerimientos funcionales y requerimientos no funcionales, que serán verificados:

- Requerimientos Funcionales:
 - Registro de Usuario
 - Jugar
 - Retomar Partida
 - Ver Ranking
 - Invitar Amigos
 - Cambiar Idioma
 - Desafío 1: Sacar una foto a un elemento
 - Desafío 2: Laberinto
 - Desafío 3: Reproducir patrón en pantalla
 - Desafío 4: Búsqueda de código QR
 - Desafío 5: Orientar Telefono
 - Desafío 6: Veo Veo
 - Desafío 7: Simon dice
 - Desafío 8: Carrera
- Requerimientos No Funcionales:
 - Performance: Tiempo de Respuesta no mayor a 3 (tres) segundos por cada operación.
 - Interfaz de usuario: debe respetar lineamientos de interfaz nativa.
 - Detección de Idioma
 - Conectividad del Cliente
 - Sistema incremental de Puntuación
 - Cantidad de desafíos
 - Puntuación independiente del día

Al estar aún en proceso de relevamiento de requerimientos, esta lista podrá sufrir modificaciones en futuras iteraciones.

3. Estrategia de Verificación

3.1. Tipos de pruebas

3.1.1. Prueba de integridad de los datos y la base de datos

3.1.1.1. Objetivo de la prueba

Asegurar que los métodos y procesos de acceso a la base de datos funcionan correctamente y sin corromper datos.

3.1.1.2. Técnica

Se utilizarán pruebas de partición para cada método o proceso utilizado para acceso a la base de datos, verificando que los datos se carguen correctamente en la base de datos y que las consultas retornan los datos correctos.

3.1.1.3. Criterio de aceptación

Todos los métodos y procesos que tengan acceso a la base de datos deberán funcionar tal y como fueron diseñados y no deberá producirse corrupción de datos.

3.1.1.4. Consideraciones especiales

Las pruebas sobre la base de datos se realizarán de forma manual, así como sus controles, y requerirán de acceso y permisos para operar en la misma.

Se deben proveer bases de datos pequeñas para aumentar la facilidad de inspección de los datos y verificar que no sucedan eventos no aceptables.

3.1.2. Prueba de Funcionalidad

3.1.2.1. Objetivo de la prueba

Asegurar que la funcionalidad del objetivo de la prueba (caso de uso, función, regla de negocio, etc.) sea la apropiada. Esto incluye tanto la navegación o el ingreso de datos, como el proceso de recuperación frente a fallos.

3.1.2.2. Técnica

Se generarán pruebas unitarias para cada caso de uso (o flujo), buscando verificar que el resultado obtenido equivale al esperado.

Las pruebas frente a casos de uso nuevos serán manuales, mientras que, una vez validado el caso de uso, se desarrollarán pruebas automatizadas que validarán la correctitud de la ejecución.

Estas pruebas automatizadas resultan de extrema utilidad para comprobar que casos de uso mantengan su correcto funcionamiento frente a cambios introducidos.

3.1.2.3. Criterio de aceptación

Todas las pruebas planificadas se realizaron. Todos los defectos encontrados han sido debidamente identificados, documentados y reportados.

3.1.2.4. Consideraciones especiales

Se deberá tener acceso a la información de cambios que se introduzcan en una nueva versión, así como de los bugs solucionados.

También será necesario contar con la documentación correspondiente a los casos de uso o flujos a probar, para poder verificar su funcionamiento.

En caso de ser necesario, se deberá contar con datos (o medios para crear dichos datos) necesarios para la ejecución de las pruebas funcionales.

3.1.3. Prueba de Ciclo del Negocio

3.1.3.1. Objetivo de la prueba

El objetivo de esta prueba es asegurar que la aplicación funciona de acuerdo a los requerimientos establecidos del ciclo de negocio. En particular, será de utilidad comprobar que el funcionamiento del ciclo completo de una partida semanal sea el correcto.

3.1.3.2. Técnica

La prueba consistirá en simular diversos ciclos de negocio, modificando los parámetros de la prueba y simulando varios usuarios diferentes de manera "simultánea".

Se verificará que el ciclo de negocio se ejecute correctamente y que los resultados que se obtengan sean válidos, habiendo aplicado correctamente cada una de las reglas de negocio.

3.1.3.3. Criterio de aceptación

Todas las pruebas planificadas se realizaron. Todos los defectos encontrados han sido debidamente identificados, documentados y reportados.

3.1.3.4. Consideraciones especiales

Será necesario soporte técnico con el manejo de fechas para poder simular de manera rápida y sin errores ciclos de negocio, cuya validación en tiempo real tomaría demasiado.

3.1.4. Prueba de Interfase de Usuario

3.1.4.1. Objetivo de la prueba

Verificar que la navegación a través de los elementos de la interfaz de usuario reflejen el comportamiento esperado descrito por los analistas. También se aprovechará para verificar que las interfaces de usuario cumplan con los estándares indicados para cada plataforma.

3.1.4.2. Técnica

Para verificar la navegación se implementarán mayoritariamente pruebas funcionales automatizadas que comprobarán que los elementos de la interfaz de usuario se comporten de acuerdo al comportamiento esperado. En casos donde sea imposible (o poco aconsejable) realizar pruebas automatizadas se diseñarán casos de prueba manuales.

La verificación de cumplimiento de los estándares de diseño por parte de la interfaz se realizará mediante pruebas manuales.

3.1.4.3. Criterio de aceptación

Todos los elementos de cada una de las interfaces de usuario, así como todos sus posibles comportamientos, son verificados de manera exitosa.

3.1.4.4. Consideraciones especiales

Ninguna.

3.1.5. Prueba de Performance

3.1.5.1. Objetivo de la prueba

Verificar que se cumplan los requerimientos de Performance establecidos con el cliente (ej: tiempos de ejecución)

3.1.5.2. Técnica

En caso de ser posible, se desarrollarán pruebas automatizadas de performance que simulen diversos flujos y calculen los tiempos de espera generados tras cada instrucción. Estas pruebas son dependientes del tráfico generado, por lo que se buscará parametrizar el tráfico generado de manera incremental, aumentando la cantidad de usuarios que ejecutan simultáneamente. También será importante añadir dentro de los parámetros de la prueba las diferencias entre las distintas plataformas sobre las que correrá la aplicación móvil, en particular probar tiempos de respuesta con baja tasa de transferencia de datos.

De no ser posible el uso de pruebas automatizadas, se procederá a realizar las mismas mediante verificaciones manuales.

3.1.5.3. Criterio de aceptación

Cumplimiento de los requerimientos de performance establecidos

3.1.5.4. Consideraciones especiales

La prueba de performance se debe realizar en una máquina dedicada para permitir control total y medición exacta. Las bases de datos usadas para las pruebas de performance deben tener un tamaño similar a las reales para mejor simulación de los tiempos de búsqueda.

Contar con las herramientas adecuadas de software y/o hardware para poder realizar las pruebas y las mediciones.

3.1.6. Prueba de Carga

3.1.6.1. Objetivo de la prueba

Verificar el comportamiento de performance de determinados componentes del software bajo diferentes condiciones de trabajo. El objetivo es determinar y asegurar que el sistema funciona apropiadamente en circunstancias de máxima carga de trabajo esperada.

3.1.6.2. Técnica

Se utilizarán pruebas Funcionales o de Ciclo de Negocio previamente desarrolladas, buscando mediante la modificación de las mismas aumentar el número de transacciones que se realizan.

3.1.6.3. Criterio de aceptación

Aprobación exitosa de las pruebas para escenarios de múltiples transacciones y/o múltiples usuarios.

3.1.6.4. Consideraciones especiales

Servidor dedicado para poder tener control de las cargas que se generan y, gracias a ello, mayor exactitud en las mediciones.

Requerirá de Bases de Datos de tamaños similares a los esperados.

3.1.7. Prueba de Esfuerzo (stress, competencia por recursos, bajos recursos)

3.1.7.1. Objetivo de la prueba

Verificar que el software funciona apropiadamente y sin error bajo condiciones de esfuerzo, como son:

- Sobrecarga del procesador en los dispositivos móviles.
- Alta tasa de clientes conectados (servidor).
- Alta tasa de pedidos simultáneos (servidor).

Será de sumo interés al realizar estas pruebas poder identificar y documentar los escenarios límites para los que el comportamiento de la aplicación permanece dentro de los límites establecidos para catalogarlo como aceptable.

3.1.7.2. Técnica

Se utilizarán pruebas automatizadas para forzar escenarios críticos que servirán para probar cada una de las limitantes que puedan surgir.

Las cargas se irán aumentando incrementalmente hasta lograr encontrar el “punto de inflexión” donde los errores comiencen a surgir (de existir dicho punto).

3.1.7.3. Criterio de aceptación

La totalidad de las pruebas planeadas ejecuta correctamente y los valores límite alcanzados están dentro de los límites satisfactorios de funcionamiento.

3.1.7.4. Consideraciones especiales

Se requerirán herramientas externas y un entorno de trabajo que pueda variar sus especificaciones técnicas.

3.1.8. Prueba de Volumen

3.1.8.1. Objetivo de la prueba

Someter al servidor a grandes demandas de acceso/consulta a la base de datos, así como también someter a la base de datos a cantidades de datos elevados en poco tiempo.

3.1.8.2. Técnica

Pruebas de Performance y Pruebas de Carga con alta tasa de accesos a la Base de Datos y/o tamaño de base de datos elevado.

3.1.8.3. Criterio de aceptación

Todas las pruebas planificadas se ejecutaron y los resultados alcanzados son satisfactorios

3.1.8.4. Consideraciones especiales

Entornos especiales de Servidor para realizar las pruebas.

3.1.9. Prueba de Seguridad y Control de Acceso

3.1.9.1. Objetivo de la prueba

Verificar que un actor pueda acceder solo a las funciones o datos para los que posee permisos

3.1.9.2. Técnica

Realizar un relevamiento de los tipos de usuario existentes y aquellos datos y funciones a los que tiene permiso.

Para cada tipo de usuario, se crearán pruebas (automáticas y/o manuales) para verificar que haya acceso a lo que tienen permitido y el acceso sea denegado para funciones o datos para los que no posee autorización.

3.1.9.3. Criterio de aceptación

Para cada tipo de actor existente en el sistema, se registra una respuesta correcta al intentar acceso a datos o funciones. Dicha respuesta dependerá de los permisos de cada tipo de usuario.

3.1.9.4. Consideraciones especiales

Documentación referente a tipos de actores del sistema y sus permisos.

3.1.10. Prueba de Fallas y Recuperación

3.1.10.1. Objetivo de la prueba

El objetivo de las pruebas de Fallas y Recuperación es someter al sistema a condiciones extremas que provoquen fallas en el mismo, para verificar que la rutina de recuperación ante dicha falla es correcta y no genera indisponibilidad ni corrupción de datos.

3.1.10.2. Técnica

Utilizando las pruebas generadas para Funcionalidad y Ciclos de Negocio y teniendo seguridad de que el funcionamiento del sistema frente a ellas es el esperado, se procederá a generar la mayor gama de fallos críticos posibles en distintos puntos de las pruebas y se verificará que el sistema se recupere de manera adecuada.

Pérdidas de energía de los dispositivos móviles o el servidor y fallos de conexión, corrupción de la base de datos son algunos de los errores a los que se someterá el software.

Estas pruebas se espera realizarlas exclusivamente de forma manual.

3.1.10.3. Criterio de aceptación

Para todos los casos de errores inducidos, tanto la aplicación móvil, como el servidor central (incluyendo la base de datos) deberán haber reaccionado frente a ellos y ejecutado correctamente las rutinas de atención frente a estos errores para haber reestablecido a un estado conocido y deseable.

3.1.10.4. Consideraciones especiales

Se requerirá de hardware dedicado para la prueba, para poder tener control frente a las fallas generadas y el estado a recuperar. Además se deberá contar con accesos especiales para poder verificar la correctitud de la recuperación, especialmente a nivel de base de datos.

Se estudiarán las posibilidades para simular estas pruebas y no recurrir a procesos "brutos" como desconectar cables.

3.1.11. Prueba de Configuración

3.1.11.1. Objetivo de la prueba

Verificar que el Software desarrollado funcione frente a configuraciones de software y hardware mínimas establecidas.

3.1.11.2. Técnica

Utilizando las Pruebas Funcionales previamente desarrolladas, configurar el sistema para simular distintas configuraciones de hardware y software y verificar que las pruebas sigan funcionando correctamente

3.1.11.3. Criterio de aceptación

Todas las pruebas planificadas deben correr correctamente para cada una de las configuraciones distintas que se decida probar.

3.1.11.4. Consideraciones especiales

Documentación de las diversas configuraciones sobre las que se realizan pruebas.

Posibilidad de simular cada una de las configuraciones planificadas en las pruebas.

3.1.12. Prueba de Instalación

3.1.12.1. Objetivo de la prueba

Las pruebas de instalación se dividen en tres objetivos clave:

- Verificar que la instalación del software se realiza correctamente para diversas configuraciones de hardware y software.
- Verificar que las actualizaciones de software se realizan correctamente en las diversas configuraciones de hardware y software.
- Verificar que, luego de la instalación, el funcionamiento de la aplicación es el esperado.

3.1.12.2. Técnica

Pruebas manuales en diversas configuraciones. Instalación y actualización.

Uso de pruebas funcionales ya implementadas para probar el funcionamiento correcto de la aplicación instalada/actualizada.

3.1.12.3. Criterio de aceptación

La instalación/actualización es exitosa y el producto instalado supera las pruebas funcionales.

3.1.12.4. Consideraciones especiales

Necesidad de poder simular diversas configuraciones para instalación/actualización.

3.1.13. Prueba de Documentos

3.1.13.1. Objetivo de la prueba

Verificar correctitud y consistencia de la documentación generada, así como garantizar que la misma sea entendible para el receptor de la misma.

3.1.13.2. Técnica

Verificación manual del documento. Desarrollo de pautas para la generación de documentos.

3.1.13.3. Criterio de aceptación

El documento generado es entendible, correcto y consistente.

3.1.13.4. Consideraciones especiales

Ninguna.

3.2. Herramientas

Herramientas a utilizar:

- Gestión de Proyecto:
 -
- Seguimiento de errores:
 - Redmine
- Monitoreo de cubrimiento de pruebas:
 -
- Automatización de Pruebas para Android:
 - Se está estudiando el uso de herramientas de software libre como:
 - Appium (<http://appium.io/>).
 - Calabash (<http://calaba.sh/>).
- Automatización de Pruebas para Windows Phone:
 - Se recomendó por parte del Cliente el uso de Microsoft Test, Visual Studio.

Muchas de las decisiones en cuanto a herramientas externas a utilizar aún no ha sido definida dentro del grupo.

4. Recursos

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

4.1. Roles

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

Rol	Cantidad mínima de recursos recomendada	Responsabilidades
Responsable de verificación	1	<ul style="list-style-type: none">• Identificar, priorizar e implementar los casos de prueba.• Armar el Plan de Verificación.• Generar el Modelo de Prueba.• Evaluar el esfuerzo necesario para realizar la verificación.• Administración de recursos necesarios.• Revisión de resultados obtenidos.• Comunicación de resultados.
Asistente de verificación	3	<ul style="list-style-type: none">• Ejecución de Pruebas.• Registro y documentación de resultados.• Recuperación del software ante errores.
Administrador de Base de Datos	1	<ul style="list-style-type: none">• Realiza la gestión y el mantenimiento del entorno de los datos de prueba y los recursos.• Administra la base de datos de Prueba.

4.2. Sistema

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

Recurso	Nombre/Tipo
Servidor de base de datos	No definido
Red o subred	No definido
Nombre del servidor	No definido
Nombre de la base de datos	No definido
PC Cliente para pruebas	No definido
Requerimientos especiales	No definido
Repositorio de pruebas	No definido
Red o subred	No definido
Nombre del servidor	No definido

5. Hitos del proyecto de Verificación

Actividad que determina el hito	Esfuerzo	Fecha de comienzo	Fecha de finalización
Planificar la verificación	10 hrs.		
Fase I - Iteración II Semanas 3, 4		01/09/2014	14/09/2014
Planificar casos de prueba	4 hrs.		
Planificar la verificación de iteración	4 hrs.		
Ajuste y Control de Verificación	10 hrs.		
Fase II - Iteración I Semanas 5, 6		15/09/2014	28/09/2014
Planificar Casos de prueba	5 hrs.		
Diseñar Casos de Prueba	15 hrs.		
Ajuste y Control de Verificación	4 hrs.		
Crear y Adecuar ambiente de Verificación	8 hrs.		
Planificar la Verificación de iteración	4 hrs.		
Fase II - Iteración II Semanas 7, 8		29/09/2014	12/10/2014
Planificar casos de prueba	5 hrs.		
Diseñar Casos de Prueba	10 hrs.		
Ajuste y Control de Verificación	4 hrs.		
Planificar la Verificación de iteración	4 hrs.		
Ejecutar la Verificación	15 hrs.		

Evaluar la Verificación	8 hrs.		
Fase III - Iteración I Semanas 9, 10		13/10/2014	26/10/2014
Diseñar Casos de Prueba	10 hrs.		
Ajuste y Control de Verificación	4 hrs.		
Planificar la Verificación de iteración	4 hrs.		
Ejecutar la Verificación	25 hrs.		
Evaluar la Verificación	10 hrs.		
Fase III - Iteración II Semanas 11, 12		27/10/2014	09/11/2014
Ajuste y Control de Verificación	4 hrs.		
Planificar la Verificación de iteración	4 hrs.		
Ejecutar la Verificación	40 hrs.		
Evaluar la Verificación	15 hrs.		
Fase IV - Iteración I Semanas 13, 14		10/11/2014	23/11/2014
Ejecutar la Verificación	40 hrs.		
Evaluar la Verificación	35 hrs.		

6. Entregables

6.1. Modelo de Casos de Prueba

Documento	Modelo de Casos de Prueba
Creado por	El Responsable de Verificación, Gonzalo Sintas.
Para quien	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	La primera liberación será el 14/09/2014.

6.2. Informes de Verificación

Documento	Se genera un documento Informe de Verificación Unitaria por cada prueba unitaria que se realice al sistema.
Creado por	Las personas que ejecutan las pruebas.
Para quien	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 unitaria.

Documento	Se genera un documento Informe Consolidación por cada consolidación que se realice al sistema.
Creado por	Las personas que ejecutan las pruebas.
Para quien	Es el retorno para los implementadores de la tarea de consolidación, que detalla los errores encontrados para que puedan ser corregidos.
Fecha de liberación	Será liberado luego de cada consolidación.

Documento	Se genera un documento Informe de Verificación de Integración por cada prueba de integración que se realice al sistema.
Creado por	Las personas que ejecutan las pruebas.
Para quien	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 integración.

Documento	Se genera un documento Informe de Verificación de Sistema por cada prueba de sistema que se realice.
Creado por	Las personas que ejecutan las pruebas.
Para quien	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.

6.3. 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 los Informes de verificación.
Para quien	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	Será liberado luego de cada verificación, unitaria, de integración y de sistema.

6.4. 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 quien	Indica el estado del sistema.
Fecha de liberación	Será liberado luego de la verificación final del sistema.

7. Dependencias

7.1. Dependencia de personal

Se requerirá de un equipo conformado por un mínimo de cinco personas, con conocimiento de:

- Programación y ejecución de pruebas automatizadas.
- Conocimientos de Bases de Datos.

Estos conocimientos no son necesarios para cada uno de los integrantes, se formarán “especialistas” para cada área de conocimiento con el fin de agilizar la generación de las pruebas.

Además será necesario poder contar con disponibilidad del Grupo de Implementación para obtener soporte en caso de ser necesario.

7.2. Dependencia de software

Dependiendo del conjunto de pruebas a ejecutar, el software a verificar deberá haber superado las pruebas de desarrollo realizadas por el grupo de implementación que garanticen la productividad de las pruebas a realizar. Ambientes que presentan errores críticos ya conocidos es muy probable que no generen resultados útiles en pruebas posteriores.

Se necesitará también contar con emuladores para Windows Phone y Android para las versiones de Sistema Operativo solicitadas y que cuenten con la posibilidad de emular todas las operaciones necesarias para la Verificación y Validación.

7.3. Dependencia de hardware

Para pruebas que lo necesiten, será de utilidad poder contar con un servidor dedicado para garantizar la correctitud en la ejecución de las mismas.

Será de utilidad poder contar con dispositivos móviles con sistemas operativos Windows Phone y Android (de versiones acordes a las sugeridas para la implementación) para poder desarrollar prueba sobre los mismos.

7.4. Dependencia de datos y base de datos de prueba

Se deberá proveer al grupo de pruebas con los datos necesarios para realizar las pruebas (o en su defecto, mecanismos para conseguirlos).

A nivel de base de datos, será necesario proveer de mecanismos de acceso y modificación de la misma en un ambiente dedicado para la realización de ciertas pruebas.

8. Riesgos

8.1. Planificación

Una mala planificación del proyecto, que abarca una mala estimación del esfuerzo necesario para realizar distintas tareas, una mala distribución de las mismas en el tiempo y una incorrecta asignación de cada una, podría implicar no solo riesgos de atraso a nivel del desarrollo del proyecto, sino que este atraso mencionado se trasladaría directamente a la planificación de Verificación y Validación realizada, acortando plazos y generando la necesidad de un sobreesfuerzo de parte del grupo encargado de la verificación.

Para mitigar los riesgos frente a incumplimiento de plazos en los tiempos de verificación necesarios, se hizo una sobreestimación de los tiempos necesarios para cada tarea, para poder contar con un extra de tiempo en caso de demoras.

A su vez, se dispuso no destinar el total de horas, que se espera tendrá cada integrante del equipo a las tareas de verificación semanalmente, en el cálculo de los tiempos de trabajo necesario para ejecutar las pruebas, con la intención de generar

un margen de error en las horas de trabajo para atajar posibles errores de estimación que inicialmente pueden existir.

8.2. Técnico

La necesidad de crear dos aplicaciones móviles representa un riesgo claro a la hora de realizar la verificación, ya que será necesario un buen trabajo de coordinación y planificación para poder cumplir con el alcance total del Plan de Verificación y Validación para ambos sistemas.

No poder contar con emuladores de Windows Phone de fácil acceso y que permitan realizar pruebas para todas las funcionalidades, ni con dispositivos móviles con este sistema operativo, hará que las pruebas que se deben realizar sobre este sistema probablemente requiera de un mayor esfuerzo de parte del equipo. Será de vital importancia poder contar con un dispositivo móvil para prueba, de lo contrario se requerirá reevaluar los tiempos que se dedicarán para las pruebas de la aplicación sobre Windows Phone.

8.3. Gestión

El relacionamiento, tanto dentro del grupo de verificación como con las demás áreas del proyecto, será clave para garantizar la correcta ejecución de las tareas de verificación y validación, especialmente tomando en cuenta que el grueso de las tareas de verificación y validación se realizarán sobre el final del proyecto, cuando el desgaste del mismo y las presiones por la fecha límite de entrega se empiezan a sentir.

9. Apéndice

9.1. Niveles de gravedad de error

En muchas actividades del proceso de verificación se deben clasificar los errores según su nivel de gravedad. Se asigna un nivel de gravedad a los errores para poder capturar de alguna manera su impacto en el sistema. Además para poder evaluar la verificación y el sistema.

A continuación se presentan los cuatro niveles diferentes de gravedad de error:

- **Catastrófico:** un error cuya presencia impide el uso del sistema.
- **Crítico:** un error cuya presencia causa la pérdida de una funcionalidad crítica del sistema. Si no se corrige el sistema no satisfará las necesidades del cliente.

- **Marginal:** un error que causa un daño menor, produciendo pérdida de efectividad, pérdida de disponibilidad o degradación de una funcionalidad que no se realiza fácilmente de otra manera.
- **Menor:** un error que no causa perjuicio al sistema, pero que requiere mantenimiento o reparación. No causa pérdida de funcionalidades que no se puedan realizar de otra manera.

9.2. Niveles de aceptación para lo elementos verificados

Niveles de aceptación:

- **No Aprobado:** el elemento verificado presenta errores catastróficos que impiden su uso o tiene errores críticos que hacen que el elemento verificado no sea confiable.
- **Con Observaciones:** El elemento verificado no presenta errores catastróficos ni errores críticos, pero si presenta errores marginales que hacen que el elemento de software no cumpla con lo especificado en ciertas circunstancias.
- **Aprobado:** el elemento verificado no tiene errores o presenta errores menores que no afectan el normal funcionamiento del elemento