

# CONNECT! & WHERE IS MY FRIEND? PLAN DE VERIFICACIÓN Y VALIDACIÓN VERSIÓN 1.6

## Historia de revisiones

Fecha	Versión	Descripción	Autor
29/08/2013	1.0	Creación del documento.	Victor Díaz
06/09/2013	1.1	Completar puntos pendientes	Victor Díaz
15/09/2013	1.2	Completar puntos pendientes	Victor Díaz
21/09/2013	1.3	Completar puntos pendientes	Victor Díaz
02/10/2013	1.4	Ajustes por cambios de fecha	Victor Díaz
13/10/2013	1.5	Agregados	Victor Díaz
1/11/2013	1.6	Revisión final SQA-Look&Feel	Leonardo Clavijo

# CONTENIDO

<b>1.INTRODUCCIÓN.....</b>	<b>3</b>
<b>1.1.PROPÓSITO</b>	<b>3</b>
1.1.1.Componenetes a Verificar.....	3
1.1.2.Pruebas.....	4
1.1.3.Recursos.....	4
1.1.4.Entregables.....	4
<b>1.2.PUNTO DE PARTIDA</b>	<b>4</b>
<b>1.3.ALCANCE</b>	<b>5</b>
1.3.1.Pruebas unitarias:.....	5
1.3.2.Pruebas de integración:.....	5
1.3.3.Pruebas exploratorias:.....	6
1.3.4.Pruebas de casos de uso:.....	6
1.3.5.Pruebas de funcionalidad:.....	6
1.3.6.Pruebas de usabilidad:.....	6
1.3.7.Pruebas de instalación:.....	7
<b>1.4.IDENTIFICACIÓN DEL PROYECTO</b>	<b>7</b>
<b>1.5.ESTRATEGIA DE EVOLUCIÓN DEL PLAN</b>	<b>7</b>
<b>2.REQUERIMIENTOS PARA VERIFICAR.....</b>	<b>8</b>
<b>3.ESTRATEGIA DE VERIFICACIÓN.....</b>	<b>10</b>
<b>3.1.TIPOS DE PRUEBAS</b>	<b>10</b>
3.1.1.Prueba de Funcionalidad.....	10
3.1.2.Prueba de Usabilidad.....	11
3.1.3.Prueba de Instalación.....	11
<b>3.2.HERRAMIENTAS</b>	<b>12</b>
<b>4.RECURSOS.....</b>	<b>13</b>
<b>4.1.ROLES</b>	<b>13</b>
<b>5.HITOS DEL PROYECTO DE VERIFICACIÓN.....</b>	<b>15</b>
<b>6.ENTREGABLES.....</b>	<b>17</b>
<b>6.1.MODELO DE CASOS DE PRUEBA</b>	<b>17</b>
<b>6.2.INFORMES DE VERIFICACIÓN</b>	<b>17</b>
<b>6.3.EVALUACIÓN DE LA VERIFICACIÓN</b>	<b>18</b>
<b>6.4.INFORME FINAL DE VERIFICACIÓN</b>	<b>18</b>
<b>7.APÉNDICE.....</b>	<b>19</b>
<b>7.1.NIVELES DE GRAVEDAD DE ERROR</b>	<b>19</b>

## 1. INTRODUCCIÓN

### 1.1. PROPÓSITO

Este Plan de Verificación para el proyecto "Connect! – Where is my Friend?" soporta los siguientes objetivos:

#### 1.1.1. COMPONENTES A VERIFICAR

Los componentes de software a verificar serán ambos productos "Connect!" y "Where is my Friend?" mediante los prototipos que se vayan generando en el transcurso del proyecto. Sin dejar de lado la realización por parte de los desarrolladores de las correspondientes pruebas de verificación unitarias y de integración sobre los diversos artefactos de software generados.

Los requerimientos a verificar dentro del sistema Connect! serán:

- Registrar usuario.
- Ingreso de permisos.
- Editar Usuario.
- Generar Datos.
- Leer Datos.
- Conectar.
- Listar Contactos.
- Eliminar Contacto.
- Sincronizar Contactos.
- Ver Pendientes.

Los requerimientos a verificar dentro del sistema Where is my Friend?:

- Listar usuarios.
- Envío de Solicitud.
- Aceptar Solicitud.
- Ver ubicación.
- Cancelar Solicitud GPS.
- Cancelar Seguimiento GPS.
- Los requerimientos no funcionales del sistema integro serán:
- Interfaz Gráfica – Atractiva- Amigable para el usuario.
- Mantenibilidad.
- Facilidad de Uso.
- Funcionabilidad.
- Escalabilidad.
- Seguridad (Where is my Friend?).
- Deploy en Azure.

---

### **1.1.2. PRUEBAS**

Por las características de ambos sistemas, la diversidad de plataformas y contextos se recomienda centrar la verificación en cada paso del proceso de desarrollo ya sea con las pruebas unitarias (del tipo de chequeos manuales, de caja blanca o de caja negra) o de integración, así como también con prototipos que se vayan generando. A estos últimos se les realizara un testing exploratorio, aprobado dicha verificación se procederá a aplicarle una verificación planificada en base a verificaciones por casos de usos. También se realizaran pruebas de funcionalidad, usabilidad e instalación.

---

### **1.1.3. RECURSOS**

Los recursos necesarios para llevar a cabo la verificación estarán dados por:

- Gimena Bernadet - Asistente de Verificación.
- Dario Britos - Asistente de Verificación.
- Valentina Da Silva - Asistente de Verificación.
- Leonardo Clavijo - Asistente de Verificación.
- Victor Diaz - Responsable de Verificación.
- Desarrolladores, para el caso de pruebas unitarias y de integración.

A los antes mencionados se les brindara una inducción acerca de las diversas tareas a realizar, así como también todos los documentos y guías básicas para llevarlas a cabo.

---

### **1.1.4. ENTREGABLES**

Los entregables a realizar serán (código del documento):

- Plan de Verificación y Validación (VRPVV).
- Plan de Verificación de la Iteración (VRPVI).
- Modelo de Casos de Prueba (VRMCP).
- Informes de Verificación (VRIVI y VRIVS).
- Informes de Verificación de Documentos (VRIVD).
- Evaluación de la verificación (VREV).
- Informe final de verificación (VRIFV).

## **1.2. PUNTO DE PARTIDA**

Para el proyecto se ha decidido realizar una verificación y validación del producto y sus diversos componentes, tal que la misma garantice que el sistema final a entregar cumple con todos los requerimientos y expectativas por parte del cliente, además de también de garantizar que el proyecto ha sido pasado por controles estrictos y rigurosos que aseguran su correcto desarrollo y desempeño. Todo esto con el fin de entregar un producto con la calidad debida y asegurada por el proceso de elaboración del mismo.

Los principales componentes a verificar y validar serán el sistema Connect!, Where is my Friend? y el servidor encargado de mediar entre sistemas y usuarios y sus interacciones.

De acuerdo a lo relevado con el cliente se destaca el interés por que los sistemas sean verificados y validados con:

- Pruebas unitarias.
- Pruebas de integración.
- Pruebas exploratorias.
- Pruebas de casos de uso.
- Pruebas de funcionalidad.
- Pruebas de usabilidad.
- Pruebas de instalación.

Cabe destacar que en cada iteración, prototipo, artefacto o etc. que pase por la etapa de verificación así corresponda, las pruebas se realizaran en el orden indicado, pudiéndose omitir alguna acorde a las características del objeto a verificar, así como también en el caso que el objeto no cumpla con los estándares mínimos establecidos se devolverá al equipo de desarrollo junto con los informes correspondientes para que estos solucionen los puntos indicados. Dado este último caso se retomaran la verificación desde la primera prueba cotejando resultados anteriores para descartar inserción de nuevas faltas o fallas.

### **1.3. ALCANCE**

En esta sección se describirán las fases o estados de la verificación, tipos de prueba, que características serán objeto de prueba y cuáles no, supuestos que pueden impactar en el diseño, desarrollo o implementación de la verificación, se enumeraran riesgos y contingencias que puedan afectar el diseño, desarrollo o implementación de la verificación, se enumeraran restricciones que pueda afectar el diseño, desarrollo o implementación de la verificación.

---

#### **1.3.1. PRUEBAS UNITARIAS:**

Las realizaran los desarrolladores con el fin de asegurar el correcto funcionamiento de cada módulo por separado. Las mismas podrán ser inspección de código (chequeos sintácticos y semánticos), pruebas de caja blanca (si fueran factibles, basados en los casos de uso y teniendo en cuenta puntos de interés del desarrollador) o pruebas de caja negra (podrán basarse en las diversas técnicas de la verificación planificada). Esto se ajustara a las características de cada modulo, desarrollo y etapa del proyecto. Para estas pruebas el objetivo es el modulo sin especular con la interacción con otros módulos, por lo cual la intención será verificar sus principales características. Dentro de los riesgos se pueden destacar el dominio del problema y el ambiente de ejecución, para lo cual por las características dadas se podrán contemplar cuando se den pruebas de integración y tanto el dominio y ambiente sean más aproximados a la realidad del sistema. Respecto a las restricciones podrán estar dadas por el volumen del desarrollo y los tiempos estipulados para cumplir las diferentes entregas, además de los recursos asignados, por lo cual se deberá seleccionar adecuadamente que verificar y de qué forma, con el fin de obtener los mejores resultados.

---

#### **1.3.2. PRUEBAS DE INTEGRACIÓN:**

Las realizaran los desarrolladores con el fin de asegurar el correcto funcionamiento de cada módulo con respecto de los restantes módulos con los cuales debe tener interacción. Para las mismas se aplicaran de forma análoga las técnicas mencionadas en las pruebas unitarias, con el objetivo de afirmar lo ya verificado y

complementar las actividades que por las diferentes restricciones y riesgos no hayan sido posibles de verificar en el punto anterior. En estas pruebas se tendrán los mismos riesgos y restricciones del punto anterior.

---

### **1.3.3. PRUEBAS EXPLORATORIAS:**

Cumplidas en caso que correspondan las pruebas anteriores se pasaran a aplicarle por parte del grupo de verificación las diversas pruebas exploratorias al artefacto en cuestión, con el fin de comprobar que dicho artefacto cumple con cierto grado de calidad que asegura que el resultado a obtener en las subsiguientes pruebas sea coherente y de utilidad, en caso negativo con el fin de maximizar y optimizar tiempos y recursos el artefacto será devuelto a la etapa de desarrollo y rechazado por parte del grupo de verificación, todo esto con la debida información inherente para la solución de los problemas detectados, subsanado esto se realizaran las mismas pruebas para corroborar su solución así como también otros conjuntos de pruebas para asegurar que en la solución de estos problemas no se hayan inyectado nuevas fallas o faltas. Para realizar estas pruebas se elaboraran misiones las cuales se basaran en los diversos casos de uso, intereses del cliente y puntos interesantes para el grupo de verificación todos estos acorde a la realidad del artefacto a verificar. Para estas pruebas los riesgos serán los mismos que se han mencionado previamente, adicionando el riesgo en fallas de comunicación entre las diversos recursos del proyecto, para atenuar esto o subsanarlo se utilizara la debida documentación además de incrementar en esta etapa la comunicación entre ambas partes, sin que esto pueda introducir otros riesgos dada la separación de tareas entre los integrantes, dejando en claro que no se evalúa al desarrollador si no al artefacto y al proceso. En estas pruebas las restricciones siguen siendo las mismas que en las pruebas anteriores.

---

### **1.3.4. PRUEBAS DE CASOS DE USO:**

Realizadas las pruebas del punto anterior y logrando un nivel de aceptación por parte del grupo de verificación se procederá a realizar las pruebas panificadas de caso de uso por parte del grupo de verificación, estas se basaran en el universo de casos de usos presentados por el cliente y los resultantes del análisis del proyecto teniendo en cuenta según corresponda la combinación de entras posibles para así definir el tipo de prueba para los casos de uso, a modo de ejemplos se podrán utilizar técnicas de árbol de decisión, combinación por pares, maquinas d estado o etc. En primera instancia se tendrán como objetivo verificar todos los casos de uso, de acuerdo a lo que se desprenda del avance del proyecto y sus artefactos se priorizaran y se verificaran según corresponda de acuerdo a las prioridades y la etapa del proyecto. Los riesgos que pueden surgir en estas pruebas serán al igual que los puntos anteriores con la salvedad que llegada esta etapa de la verificación los mismos podrán ser más críticos por el consumo de recursos y tiempos en la elaboración del proceso. Respecto a las restricciones siguen siendo las mismas que en las pruebas anteriores.

---

### **1.3.5. PRUEBAS DE FUNCIONALIDAD:**

Estas pruebas serán realizadas por el grupo de verificación, las mismas se realizaran con el objetivo de ver que se cumplan de forma correcta los casos de uso y se respete la política de negocios planteada por el cliente, las mismas se desarrollaran en conjunto con las pruebas de caso de uso con el fin de maximizar los tiempos y recursos asignados, además de la relación estrecha que llevan ambos tipos de prueba. Respecto a riesgos y restricciones se mantienen incambiantes respecto a los puntos anteriores.

---

### **1.3.6. PRUEBAS DE USABILIDAD:**

Estas pruebas serán realizadas por el grupo de verificación, las mismas se realizaran con el objetivo de ver que se cumplan ciertos niveles de calidad respecto a la usabilidad de los sistemas, teniendo en cuenta ciertos criterios de acuerdo a la respuesta del sistema y en entorno del mismo, podrán tenerse en cuenta puntos

vista posicionándose del lado del usuario final y su interacción con el sistema. Estas pruebas se desarrollaran en conjunto con las pruebas de caso de uso con el fin de maximizar los tiempos y recursos asignados, además de la relación estrecha que llevan ambos tipos de prueba. Respecto a riesgos y restricciones se mantienen incambiantes respecto a los puntos anteriores.

---

### **1.3.7. PRUEBAS DE INSTALACIÓN:**

Estas pruebas serán realizadas por el grupo de verificación, las mismas se realizaran con el objetivo de ver que se cumplan ciertos niveles de calidad y sencillez a la hora del uso por parte de los usuarios finales según el artefacto del proyecto. Basándose también en el apoyo por parte del responsable SCM. Respecto a riesgos y restricciones se mantienen incambiantes respecto a los puntos anteriores.

Para todas las pruebas a realizar se considera el riesgo de dependencia con las diversas redes sociales y sus diversas características de servicios y funcionalidades brindadas, pudiéndose así tener diversas complicaciones extras además de las presentadas en esta sección, las mismas según corresponda serán detalladas a medida que se den en el transcurso del proyecto.

En caso que ameriten podrían sumarse pruebas de configuración, Seguridad y Control de Acceso, Interface de Usuario o etc, esto se deberá evaluar y aprobar por parte del grupo del proyecto acorde a las necesidades del proyecto en sí. En un principio de acuerdo a lo establecido con el cliente las mismas nos serian de mayor interés el enfoque principal esta dado por las pruebas ya planteadas más que las mismas puedan ser replicadas por el cliente en sí.

### **1.4. IDENTIFICACIÓN DEL PROYECTO**

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

- Documento de requerimientos\*.
- Modelo de casos de uso\*.
- Modelo de diseño\*.
- Descripción de la arquitectura\*.
- Planes de Verificación y Validación de proyectos anteriores (memoria organizacional).
- Presentación teórica de Verificación y Validación del curso de IIS 2013.
- Presentaciones correspondiente al curso de Taller de verificación de Software, edición 2012.

### **1.5. ESTRATEGIA DE EVOLUCIÓN DEL PLAN**

- Los responsables de monitorear el Plan de Verificación y Validación, serán los responsables de verificación y de SQA.
- La frecuencia con la que se podría modificar el Plan de Verificación y Validación, va a variar según el transcurso del proyecto, lo ideal es que no se den cambios en el mismo salvo excepciones por sucesos críticos o por el cierre o ajuste de los documentos presentados en el punto 1.4 marcados con \*.
- La evaluación y aprobación de los cambios al Plan de Verificación y Validación, estará dada por los integrantes del proyecto con intervención del director del mismo y del cliente.

- La comunicación de los cambios al Plan de Verificación y Validación, se dará mediante la forma habitual de comunicación de los integrantes del proyecto, pudiéndose generar en caso que corresponda algún otro flujo de comunicación alternativo de acuerdo al impacto del cambio acontecido.

## 2. REQUERIMIENTOS PARA VERIFICAR

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

Los requerimientos funcionales a verificar dentro del sistema Connect! serán:

- Registrar Usuario.
- Ingreso de permisos.
- Editar Usuario.
- Generar Datos.
- Leer Datos.
- Conectar.
- Listar Contactos.
- Eliminar Contacto.
- Sincronizar Contactos.
- Ver Pendientes.

Los requerimientos funcionales a verificar dentro del sistema Where is my Friend? serán:

- Listar usuarios.
- Envío de Solicitud.
- Aceptar Solicitud.
- Ver ubicación.
- Cancelar Solicitud GPS.
- Cancelar Seguimiento GPS.

Los requerimientos no funcionales del sistema integro serán:

- Interfaz Gráfica – Atractiva- Amigable para el usuario.
- Mantenibilidad.
- Facilidad de Uso (Usabilidad).
- Funcionabilidad.
- Escalabilidad.
- Seguridad (Where is my Friend?).
- Deploy en Azure.

Casos de uso serán:

- Idénticos a los requerimientos funcionales presentados anteriormente (en proceso).

De los anteriores los más importantes a ser verificados son:

- Registrar Usuario - Connect!.
- Ingreso de permisos - Connect!.
- Generar y publicar datos - Connect!
- Leer datos - Connect!.
- Aceptar conexión - Connect!.
- Envío de solicitud - Where is my Friend?.
- Aceptar solicitud - Where is my Friend?.
- Interfaz Gráfica – Atractiva- Amigable para el usuario.
- Usabilidad.

### 3. ESTRATEGIA DE VERIFICACIÓN

En esta sección se presentara el enfoque recomendado para la verificación. Se describe como se verificarán los elementos. Para cada tipo de prueba, se proporcionara una descripción de la prueba y porque será implementada y ejecutada.

Si un tipo de prueba no será implementada y ejecutada, se indicara brevemente cual es la prueba que no se implementará o ejecutará y se justificara el motivo.

Se indicarán las técnicas usadas y el criterio para saber cuando una prueba se completó (criterio de aceptación).

Las pruebas se ejecutaran usando ciertos juegos de datos que aseguren el correcto funcionamiento de la misma, ósea serán datos definidos con ciertos objetivos para poder verificar la existencia de faltas o fallas, se podrá tener mayor flexibilidad en los datos a usar en las pruebas del tipo exploratorio por las características de las mismas, debiéndose documentar según corresponda.

#### 3.1. TIPOS DE PRUEBAS

##### 3.1.1. PRUEBA DE FUNCIONALIDAD

La prueba de funcionalidad se enfoca en los requerimientos para verificar que se corresponden directamente a casos de usos o funciones y reglas del negocio relevadas en los análisis correspondientes, además de establecidas por el cliente. Los objetivos de estas pruebas son verificar la aceptación de los datos, el proceso, la recuperación y la implementación correcta de las reglas del negocio. 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. En base a los casos de uso definidos

###### 3.1.1.1. OBJETIVO DE LA PRUEBA

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

###### 3.1.1.2. TÉCNICA

Ejecute cada caso de uso, flujo de caso de uso, o función usando datos válidos y no válidos, para verificar lo siguiente:

- Se obtienen los resultados esperados cuando se usan datos válidos.
- Cuando se usan datos no válidos se despliegan los mensajes de error o advertencia apropiados.
- Se aplica apropiadamente cada regla del negocio.

###### 3.1.1.3. CRITERIO DE ACEPTACIÓN

Todas las pruebas planificadas se realizaron. Todos los defectos encontrados han sido debidamente identificados y documentados. Se cumple con los criterios establecidos por parte del cliente.

###### 3.1.1.4. CONSIDERACIONES ESPECIALES

No se detectan consideraciones especiales mas alla de las indicadas por el cliente respecto a que esta prueba es de mayor interés para el.

---

### **3.1.2. PRUEBA DE USABILIDAD**

La prueba de usabilidad se enfoca en el objetivo de ver que se cumplan ciertos niveles de calidad respecto a la usabilidad de los sistemas, teniendo en cuenta ciertos criterios de acuerdo a la respuesta del sistema y en entorno del mismo, podrán tenerse en cuenta puntos vista posicionándose del lado del usuario final y su interacción con el sistema.

---

#### **3.1.2.1. OBJETIVO DE LA PRUEBA**

Verificar la usabilidad del sistema realizando pruebas que se enfoquen en lo posible en un amplio espectro de usuarios finales, contemplando diversas consideraciones como ser a modo de ejemplo facilidad de uso para usuarios con distintas capacidades.

---

#### **3.1.2.2. TÉCNICA**

Ejecute cada caso de uso, flujo de caso de uso, o función usando datos válidos y no válidos, para verificar lo siguiente:

- Se obtienen los resultados esperados cuando se usan datos válidos y la respuesta del sistema se puede apreciar de forma sencilla sin mayores complicaciones.
- Cuando se usan datos no válidos se despliegan los mensajes de error o advertencia apropiados y la respuesta del sistema se puede apreciar de forma sencilla sin mayores complicaciones.

---

#### **3.1.2.3. CRITERIO DE ACEPTACIÓN**

Cada interacción con el sistema ha sido verificada exitosamente siendo de facilidad la interpretación de los resultados provistos por el sistema.

---

#### **3.1.2.4. CONSIDERACIONES ESPECIALES**

Para realizar estas pruebas se considera que los retornos del sistema tienen que ser capaces de ser interpretados por el mayor posible de diferentes tipos de usuarios finales. Considerando a modo de ejemplo dificultades auditivas, visuales o de diferente índole.

---

### **3.1.3. 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 configurarla e instalarla, 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.

---

#### **3.1.3.1. 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, un nuevo celular, nunca instalados previamente con los sistemas Connect! y Where is my Friend?
- actualización, un celular instalado previamente con los sistemas Connect! y Where is my Friend?, con la misma versión
- actualización, un celular instalado previamente con los sistemas Connect! y Where is my Friend?, con una versión anterior.

---

### 3.1.3.2. TÉCNICA

Esta prueba se realizara manualmente realizando la instalación y en caso de ser satisfactoria se ejecutaran un conjunto de pruebas funcionales ya implementadas para la Prueba de Funcionalidad.

---

### 3.1.3.3. CRITERIO DE ACEPTACIÓN

Las pruebas de funcionalidad de Connect! y Where is my Friend? se ejecutan exitosamente sin fallas y faltas.

---

### 3.1.3.4. CONSIDERACIONES ESPECIALES

Se tendrán en cuenta los diferentes entornos y combinaciones por las características solicitadas, a modo de ejemplo se precisara que se brinde un celular con las prestaciones que correspondan para las solicitudes realizadas, o tener dentro de lo posible entornos virtuales que reproduzcan el entorno real y considerar otros factores que afecten los resultados obtenidos.

Para las pruebas unitarias, de integración, modelo de casos de uso y exploratorias seguirán los procesos habituales que corresponden a cada tipo de prueba, serán detalladas en una futura versión del Plan de Verificación y Validación, cuando se cuente con información más concreta sobre el proyecto a medida que se finalicen otros artefactos del proyecto necesarios.

Las pruebas mencionadas en los puntos anteriores también se basaran en la definición de los casos de uso así como también en ciertos juegos de datos para llevarlas a cabo.

## 3.2. HERRAMIENTAS

Team Foundation Server 2012: Para reportar incidencias ocurridas y realizar el correspondiente seguimiento. Gracias a un acuerdo de la FING con Microsoft, disponemos de cuentas de prueba de AZURE, donde se instaló una maquina virtual con un entorno TFS de pruebas de 90 días, tiempo suficiente para desarrollar el proyecto satisfactoriamente.

- Unit test de C# para realizar pruebas unitarias en el Servidor.
- Qunit para realizar las pruebas unitarias en el núcleo del sistema.
- Visual Studio y Eclipse para desde el entorno de programación poder ejecutar los sistemas para realizar la verificación sobre W8 y Android respectivamente.
- iPhone emulador provisto por XCode en maquina virtual Mac/iOS.

**Nota:** Dadas las condiciones del sistema a desarrollar y la diversidad de entornos se ha decidido que no se realizaran pruebas automatizadas. Para el resto de las pruebas se utilizaran métodos manuales y si las tecnologías lo permiten se podrán a llegar utilizar herramientas para la cobertura de código.

## 4. RECURSOS

En esta sección se presentaran los recursos recomendados para el proyecto Connect! & Where is my Friend?, sus principales responsabilidades y su conocimiento o habilidades.

### 4.1. ROLES

En la tabla a continuación se muestra la composición del personal para el proyecto Connect! - Where is my Friend? 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 e implementa los casos de prueba. <ul style="list-style-type: none"><li>• Genera el Plan de Verificación y Validación.</li><li>• Genera el Modelo de Prueba.</li><li>• Evalúa el esfuerzo necesario para verificar.</li><li>• Proporciona la dirección técnica.</li><li>• Adquiere los recursos apropiados.</li><li>• Proporciona informes sobre la verificación.</li></ul>
Asistente de verificación	4	<ul style="list-style-type: none"><li>• Ejecuta las pruebas</li><li>• Registra los resultados de las pruebas.</li><li>• Recuperar el software de errores.</li><li>• Documenta los pedidos de cambio.</li></ul>
Implementadores	10	<ul style="list-style-type: none"><li>• Ejecutan pruebas unitarias y de integración.</li><li>• Registra los resultados de las pruebas.</li><li>• Recuperar el software de errores.</li><li>• Documenta los pedidos de cambio.</li></ul>
Administrador de Base de Datos	1	<ul style="list-style-type: none"><li>• Realiza la gestión y mantenimiento del entorno de los datos (base de datos) de prueba y los recursos.</li><li>• Administra la base de datos de prueba.</li></ul>
Administrador de Recursos	1	<ul style="list-style-type: none"><li>• Realiza la gestión y mantenimiento de los recursos.</li><li>• Administra los recursos para las pruebas.</li></ul>

**Nota:** Recursos en esta tabla hace referencia a todo aquello sobre usuarios de redes sociales y sus dependencias para la realización de pruebas.

### 1.1. Sistema

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

<b>Recurso</b>	<b>Nombre/Tipo</b>
Android: Emulador android version 4 o superior en entorno windows	No definido
WP8: Emulador provisto por VS2012 Express WP8, en maquina virtual W8 PRO	No definido
Teléfono Nokia con WP8, por fallas en el emulador.	No definido
W8: Entorno windows 8 estándar o pro (interface METRO)	No definido
IPhone: Emulador provisto por XCode en maquina virtual Mac/iOS	No definido
Backend : Esto sería un entorno REST	No definido

## 5. HITOS DEL PROYECTO DE VERIFICACIÓN

La verificación del proyecto Connect! - Where is my Friend? debe incorporar actividades de prueba para cada verificación identificada en las secciones anteriores. Se identifican los siguientes hitos del proyecto de verificación separados para comunicar los logros de estado de proyecto.

<b>Actividad que determina el hito</b>	<b>Esfuerzo</b>	<b>Fecha de comienzo</b>	<b>Fecha de finalización</b>
Planificar la verificación	20hs	29/08/2013	22/09/2013
Ajuste y Control de Verificación	-		
<b>Iteración 1 Fase Elaboración</b>			
Elaborar casos de prueba	10hs	04/09/2013	02/10/2013
Verificación	5hs por pers.	29/09/2013	13/10/2013
Informe de la Verificación	4hs	06/10/2013	17/10/2013
Evaluar la verificación	4hs	07/10/2013	18/10/2013
<b>Iteración 2 Fase Elaboración</b>			
Elaborar casos de prueba	10hs	14/10/2013	18/10/2013
Verificación	5hs por pers.	18/10/2013	22/10/2013
Informe de la Verificación	4hs	22/10/2013	22/10/2013
Evaluar la verificación	4hs	22/10/2013	23/10/2013
<b>Iteración 1 Fase Construcción</b>			
Elaborar casos de prueba	10hs	24/10/2013	27/10/2013
Verificación	5hs por pers.	01/11/2013	05/11/2013
Informe de la Verificación	4hs	05/11/2013	06/11/2013
Evaluar la verificación	4hs	06/11/2013	07/11/2013
<b>Iteración 2 Fase Construcción</b>			
Elaborar casos de prueba	10hs	09/11/2013	14/11/2013
Verificación	5hs por pers.	16/11/2013	19/11/2013
Informe de la Verificación	4hs	18/11/2013	19/11/2013
Evaluar la verificación	4hs	19/11/2013	20/11/2013
<b>Iteración 1 Fase Transición</b>			

Elaborar casos de prueba	8hs	10/11/2013	14/11/2013
Verificación	5hs por pers.	15/11/2013	19/11/2013
Informe de la Integración	4hs	15/11/2013	19/11/2013
Informe de la Verificación	4hs	15/11/2013	19/11/2013
Evaluar la verificación	4hs	19/11/2013	22/11/2013
Informe de Verificación de Integración	8hs	19/11/2013	22/11/2013
Informe final de verificación	8hs	17/11/2013	24/11/2013

## 6. ENTREGABLES

### 6.1. MODELO DE CASOS DE PRUEBA

Documento	<b>Modelo de Casos de Prueba</b>
Creado por	El Responsable de verificación, Victor Díaz.
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	Ver documentos de Plan de Verificación de la Iteración o punto 5. Considerados en puntos "Elaborar casos de prueba"

### 6.2. INFORMES DE VERIFICACIÓN

Documento	Se genera un documento <b>Informe de Verificación Unitaria</b> por cada prueba unitaria que se realice al sistema.
Creado por	Las personas que ejecutan las pruebas, implementadores.
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. Ver documentos de Plan de Verificación de la Iteración o punto 5. Considerados en puntos "Verificación"

Documento	Se genera un documento <b>Informe de Verificación de Documentos</b> para el Plan de Verificación y Validación y el Modelo de Casos de Prueba.
Creado por	Las personas que ejecutan las pruebas, asistente de Verificación.
Para quien	Es el retorno para el Responsable de Verificación, que detalla los errores encontrados para que puedan ser corregidos.
Fecha de liberación	Ver documentos de Plan de Verificación de la Iteración o punto 5. Considerado en el punto "Elaborar casos de prueba" de la primera iteración de la fase de elaboración.

Documento	Se genera un documento <b>Informe de Verificación de Integración</b> por cada prueba de integración que se realice al sistema.
Creado por	Las personas que ejecutan las pruebas, implementadores.
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	Ver documentos de Plan de Verificación de la Iteración o punto 5.

Documento	Se genera un documento <b>Informe de Verificación de Sistema</b> 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. Ver documentos de Plan de Verificación de la Iteración o punto 5.

### 6.3. EVALUACIÓN DE LA VERIFICACIÓN

Documento	Se genera un documento <b>Evaluación de la verificación</b> 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. Ver documentos de Plan de Verificación de la Iteración o punto 5.

### 6.4. INFORME FINAL DE VERIFICACIÓN

Documento	El documento <b>Informe final de verificación</b> 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	Ver documentos de Plan de Verificación de la Iteración o punto 5.

## 7. APÉNDICE

### 7.1. 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 da una sugerencia de 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.

### 1.2. Niveles de aceptación para lo elementos verificados

Se debe establecer un nivel de aceptación para los elementos verificados para poder establecer el estado en el que se encuentra el proyecto.

En esta sección defina niveles de aceptación y los criterios de pertenencia a cada nivel.

Como ejemplo de niveles de aceptación:

- **No aprobado:** el elemento verificado tiene errores catastróficos (uno o varios) que impiden su uso o tiene errores críticos (uno o varios) que hacen que el elemento verificado no sea confiable. El usuario no puede depender de él para realizar el trabajo.
- **Aprobado con Observaciones:** el elemento verificado no tiene errores catastróficos, ni errores críticos, pero tiene errores marginales (uno o varios) que hacen que el elemento de software se degrade en algunas situaciones.
- **Aprobado:** el elemento verificado no tiene errores o tiene errores menores que no afectan el normal funcionamiento del elemento.