

CONNECT! & WHERE IS MY FRIEND?

MODELO DE CASOS DE PRUEBA

VERSIÓN 1.1

Historia de revisiones

Fecha	Versión	Descripción	Autor
15/10/2013	1.0	Creación del documento	Victor Díaz
1/11/2013	1.1	Revisión SQA-Look&Feel	Leonardo Clavijo

Responsable de verificación: Victor Díaz.

CONTENIDO

1. CASOS Y PROCEDIMIENTOS DE PRUEBAS UNITARIAS.....	3
1.1. MAILBOXCONTROLLER EN CONNECT! VERSIÓN 1.0	3
1.1.1. Requerimientos Funcionales.....	3
1.1.2. Procedimiento de prueba.....	3
1.1.3. Resultado obtenido.....	3
1.2. MAILBOXCONTROLLER EN WHERE IS MY FRIEND? VERSIÓN 1.0	4
1.2.1. Requerimientos Funcionales.....	4
1.2.2. Procedimiento de prueba.....	4
1.2.3. Resultado obtenido.....	4
1.3. IOSPECIFICCONTROLLERQR EN CONNECT! VERSIÓN 1.0	5
1.3.1. Requerimientos Funcionales.....	5
1.3.2. Procedimiento de prueba.....	5
1.3.3. Resultado obtenido.....	5
1.4. SERVERACCESSCONTROLLER EN WHERE IS MY FRIEND? VERSIÓN 1.0	6
1.4.1. Requerimientos Funcionales.....	6
1.4.2. Procedimiento de prueba.....	6
1.4.3. Resultado obtenido.....	6
1.5. TRACKINGCONTROLLER EN WHERE IS MY FRIEND? VERSIÓN 1.0	7
1.5.1. Requerimientos Funcionales.....	7
1.5.2. Procedimiento de prueba.....	7
1.5.3. Resultado obtenido.....	7
2. CASOS Y PROCEDIMIENTOS DE PRUEBAS DEL SISTEMA.....	8
2.1. CONNECT! VERSIÓN 0.0.0.3	8
2.1.1. Requerimientos Funcionales.....	8
2.1.2. Requerimientos No funcionales.....	11
2.2. WHERE IS MY FRIEND? VERSIÓN 0.0.0.3	12
2.2.1. Requerimientos Funcionales.....	12
2.2.2. Requerimientos No funcionales.....	13

1. CASOS Y PROCEDIMIENTOS DE PRUEBAS UNITARIAS

En esta sección se detallan los Casos de Pruebas Unitarias que se realizarán y los procedimientos de prueba. En cada iteración se detallaran los Casos de Pruebas Unitarias planificados para la iteración. Para un detalle mayor de las mismas ver el documento "Informe de Verificación Unitaria (IMIVUG6)" de cada integrante mencionado.

1.1. MAILBOXCONTROLLER EN CONNECT! VERSIÓN 1.0

Implementador: Samuel Noble.

1.1.1. REQUERIMIENTOS FUNCIONALES

Lista total de las funcionalidades del componente que se deben verificar, de acuerdo al plan de verificación de la iteración. Para un detalle mayor de cada iteración ver el Plan de Verificación de la Iteración que corresponda.

Lista de funcionalidades del componente que se desean verificar de acuerdo al Plan de Verificación de la Iteración:

- *Init (Inicializar Controlador).*
- *GetMessages.*
- *GetMessage.*
- *RemoveAcceptationMessage.*
- *ResponseConnectionRequest.*

1.1.2. PROCEDIMIENTO DE PRUEBA

Dadas las dificultades de las tecnologías y la libertad ante esta problemática, las pruebas se realizaron mediante la ejecución de las funcionalidades desde el entorno de programación. Y mediante el debugger del entorno se controló la respuesta de las mismas. Para algunas de las funcionalidades. En otras si se pudo utilizar el QUnit.

1.1.3. RESULTADO OBTENIDO

Ver documento "IMIVUG6v2.0_Samuel_Noble.doc" por más detalle. A modo de resumen se destaca que se ha cubierto un 80% del código, aunque el documento refleja los resultados de un 40% del código de MailBoxController aproximadamente.

1.2. MAILBOXCONTROLLER EN WHERE IS MY FRIEND? VERSIÓN 1.0

Implementador: Samuel Noble.

1.2.1. REQUERIMIENTOS FUNCIONALES

Lista total de las funcionalidades del componente que se deben verificar, de acuerdo al plan de verificación de la iteración. Para un detalle mayor de cada iteración ver el Plan de Verificación de la Iteración que corresponda.

Lista de funcionalidades del componente que se desean verificar de acuerdo al Plan de Verificación de la Iteración:

- *Init (Inicializar Controlador).*
- *GetMessages.*
- *GetMessage.*
- *RemoveAcceptationMessage.*
- *ResponseConnectionRequest.*

1.2.2. PROCEDIMIENTO DE PRUEBA

Dadas las dificultades de las tecnologías y la libertad ante esta problemática, las pruebas se realizaron mediante la ejecución de las funcionalidades desde el entorno de programación. Y mediante el debugger del entorno se controló la respuesta de las mismas. Para algunas de las funcionalidades. En otras si se pudo utilizar el QUnit.

1.2.3. RESULTADO OBTENIDO

Ver documento "IMIVUG6v2.0_Samuel_Noble.doc" por más detalle. A modo de resumen se destaca que se ha cubierto un 80% del código, aunque el documento refleja los resultados de un 40% del código de MailBoxController aproximadamente.

1.3. IOSPECIFICCONTROLLERQR EN CONNECT! VERSIÓN 1.0

Implementador: Adrian Caretti.

1.3.1. REQUERIMIENTOS FUNCIONALES

Lista total de las funcionalidades del componente que se deben verificar, de acuerdo al plan de verificación de la iteración. Para un detalle mayor de cada iteración ver el Plan de Verificación de la Iteración que corresponda.

Lista de funcionalidades del componente que se desean verificar de acuerdo al Plan de Verificación de la Iteración:

- *publishData (Mostrar QR).*
- *startReadConnections.*

1.3.2. PROCEDIMIENTO DE PRUEBA

Dadas las dificultades de las tecnologías y la libertad ante esta problemática, las pruebas se realizaron mediante la ejecución de las funcionalidades desde el entorno de programación. Y mediante el debugger del entorno se controló la respuesta de las mismas.

1.3.3. RESULTADO OBTENIDO

Ver documento "IMIVUG6v2.0_ Adrian_Caretti.doc" por más detalle. A modo de resumen se destaca que se ha cubierto un 66% del código. Pues la funcionalidad de startReadConnections no pudo ser desarrollada.

1.4. SERVERACCESSCONTROLLER EN WHERE IS MY FRIEND? VERSIÓN 1.0

Implementador: Adrian Caretti.

1.4.1. REQUERIMIENTOS FUNCIONALES

Lista total de las funcionalidades del componente que se deben verificar, de acuerdo al plan de verificación de la iteración. Para un detalle mayor de cada iteración ver el Plan de Verificación de la Iteración que corresponda.

Lista de funcionalidades del componente que se desean verificar de acuerdo al Plan de Verificación de la Iteración

- *sendConnectionRequest (Enviar Solicitud).*
- *AcceptConnectionRequest.*

1.4.2. PROCEDIMIENTO DE PRUEBA

Dadas las dificultades de las tecnologías y la libertad ante esta problemática, las pruebas se realizarán mediante la ejecución de las funcionalidades desde el entorno de programación. Y mediante el debugger del entorno se controló la respuesta de las mismas.

Las mismas aun no han sido realizadas por falta de implementación.

1.4.3. RESULTADO OBTENIDO

Ver documento "IMIVUG6v2.0_ Adrian_Caretti.doc" por más detalle.

Las mismas aun no han sido realizadas por falta de implementación.

1.5. TRACKINGCONTROLLER EN WHERE IS MY FRIEND? VERSIÓN 1.0

Implementador: Adrian Caretti.

1.5.1. REQUERIMIENTOS FUNCIONALES

Lista total de las funcionalidades del componente que se deben verificar, de acuerdo al plan de verificación de la iteración. Para un detalle mayor de cada iteración ver el Plan de Verificación de la Iteración que corresponda.

Lista de funcionalidades del componente que se desean verificar de acuerdo al Plan de Verificación de la Iteración

- *aceptarSolicitud (Enviar Solicitud).*

1.5.2. PROCEDIMIENTO DE PRUEBA

Dadas las dificultades de las tecnologías y la libertad ante esta problemática, las pruebas se realizarán mediante la ejecución de las funcionalidades desde el entorno de programación. Y mediante el debugger del entorno se controló la respuesta de las mismas.

Las mismas aun no han sido realizadas por falta de implementación.

1.5.3. RESULTADO OBTENIDO

Ver documento "IMIVUG6v2.0_ Adrian_Caretti.doc" por más detalle.

Las mismas aun no han sido realizadas por falta de implementación.

2. CASOS Y PROCEDIMIENTOS DE PRUEBAS DEL SISTEMA

Se detallan los Casos de pruebas del Sistema que se deben verificar de acuerdo a lo establecido en el Plan de Verificación de la Iteración. Incluye la verificación de requerimientos funcionales y no funcionales del Sistema y las pruebas de interacción que correspondan.

2.1. CONNECT! VERSIÓN 0.0.0.3

Verificadores:

- Dario Britos sobre iPhone
- Gimena Bernardet sobre Windows Phone
- Valentina Da Silva sobre Android
- Victor Diaz sobre Windows 8

2.1.1. REQUERIMIENTOS FUNCIONALES

Lista total de las funcionalidades del componente que se deben verificar, de acuerdo al plan de verificación de la iteración. Para un detalle mayor de cada iteración ver el Plan de Verificación de la Iteración que corresponda.

Lista de funcionalidades del componente que se desean verificar de acuerdo al Plan de Verificación de la Iteración:

- Configurar redes sociales.
- Ingresar red social.
- Ingreso de permisos.
- Generar y Publicar Datos mediante un código QR.
- Aceptar pedido de conexión.
- Leer Datos mediante código QR.
- Sincronizar contactos.
- Listar Notificaciones.

2.1.1.1. CONFIGURAR REDES SOCIALES.

Entrada	Salida esperada
Selección de redes por default, opción de eliminarla, selección de foto y opción de ingresar nueva red social.	El sistema valida estos datos y quedan configuradas las redes a compartir.

Procedimiento de prueba

Iniciar sistema

Dirigirse a la opción de configurar redes sociales realizando cada uno los casos de pruebas indicados por vez. Ver documento: "CP It2 Fase elaboracion Connect.xls".

También se da la libertad al asistente de verificación a que agregue casos de pruebas según los resultados obtenidos durante la verificación.

Acorde a las respuestas obtenidas para cada caso de prueba registrar en documento "Reporte de Pruebas VRRPRG6v1.0.docx".

2.1.1.2. INGRESAR RED SOCIAL

Entrada	Salida esperada
Selección de red social no configurada e ingreso identificador	El sistema valida los datos y pasa al caso de uso "Ingresar Permisos".

Procedimiento de prueba

Iniciar sistema

Dirigirse a la opción de ingresar red social, realizando cada uno los casos de pruebas indicados por vez. Ver documento: "CP It2 Fase elaboracion Connect.xls".

También se da la libertad al asistente de verificación a que agregue casos de pruebas según los resultados obtenidos durante la verificación.

Acorde a las respuestas obtenidas para cada caso de prueba registrar en documento "Reporte de Pruebas VRRPRG6v1.0.docx".

2.1.1.3. INGRESO DE PERMISOS

Entrada	Salida esperada
Según Red Social	Se le brindan permisos al usuario y sistema.

Procedimiento de prueba

Iniciar sistema

Dirigirse a la opción de ingreso de permisos, realizando cada uno los casos de pruebas indicados por vez. Ver documento: "CP It2 Fase elaboracion Connect.xls".

También se da la libertad al asistente de verificación a que agregue casos de pruebas según los resultados obtenidos durante la verificación.

Acorde a las respuestas obtenidas para cada caso de prueba registrar en documento "Reporte de Pruebas VRRPRG6v1.0.docx".

2.1.1.4. GENERAR Y PUBLICAR DATOS MEDIANTE UN CÓDIGO QR

Entrada	Salida esperada
Indicación de generar.	Se publica el código QR.

Procedimiento de prueba

Iniciar sistema

Dirigirse a la opción de generar y publicar datos mediante un código QR, realizando cada uno los casos de pruebas indicados por vez. Ver documento: "CP It2 Fase elaboracion Connect.xls".

También se da la libertad al asistente de verificación a que agregue casos de pruebas según los resultados obtenidos durante la verificación.

Acorde a las respuestas obtenidas para cada caso de prueba registrar en documento "Reporte de Pruebas VRRPRG6v1.0.docx".

2.1.1.5. ACEPTAR PEDIDO DE CONEXIÓN

Nota: se deben leer datos mediante código QR previamente.

Entrada	Salida esperada
Se leen datos mediante código QR.	Los usuarios quedan conectados en las redes sociales que comparten en común.

Procedimiento de prueba

Iniciar sistema

Una vez solicitada un pedido de conexión realizar el aceptar pedido de conexión (esto es automático luego de la lectura de datos mediante código QR), realizando cada uno los casos de pruebas indicados por vez. Ver documento: "CP It2 Fase elaboracion Connect.xls".

También se da la libertad al asistente de verificación a que agregue casos de pruebas según los resultados obtenidos durante la verificación.

Acorde a las respuestas obtenidas para cada caso de prueba registrar en documento "Reporte de Pruebas VRRPRG6v1.0.docx".

2.1.1.6. LEER DATOS MEDIANTE CÓDIGO QR

Entrada	Salida esperada
Selección de leer mediante cámara.	Se dirige a la funcionalidad de aceptar pedido de conexión.

Procedimiento de prueba

Iniciar sistema

Dirigirse a la opción de leer datos mediante un código QR, realizando cada uno los casos de pruebas indicados por vez. Ver documento: "CP It2 Fase elaboracion Connect.xls".

También se da la libertad al asistente de verificación a que agregue casos de pruebas según los resultados obtenidos durante la verificación.

Acorde a las respuestas obtenidas para cada caso de prueba registrar en documento "Reporte de Pruebas VRRPRG6v1.0.docx".

2.1.1.7. SINCRONIZAR CONTACTOS

Entrada	Salida esperada
Selección de red social, selección de "todos" y usuarios a agregar.	Se agregan los usuarios seleccionados.

Procedimiento de prueba

Iniciar sistema

Dirigirse a la opción de sincronizar contactos, realizando cada uno los casos de pruebas indicados por vez. Ver documento: "CP It2 Fase elaboracion Connect.xls".

También se da la libertad al asistente de verificación a que agregue casos de pruebas según los resultados obtenidos durante la verificación.

Acorde a las respuestas obtenidas para cada caso de prueba registrar en documento "Reporte de Pruebas VRRPRG6v1.0.docx".

2.1.1.8. LISTAR NOTIFICACIONES

Nota: se deben realizar alguna solicitud antes para que se carguen las mismas.

Entrada	Salida esperada
-	Lista las notificaciones pendientes de ver.

Procedimiento de prueba

Iniciar sistema

Esperar por que se listen las notificaciones.

Para esto se podrán tener diferentes respuestas según la actividad sobre el sistema. Los resultados de estas pruebas se darán a medida de la realización de las restantes pruebas antes descritas.

También se da la libertad al asistente de verificación a que agregue casos de pruebas según los resultados obtenidos durante la verificación.

Acorde a las respuestas obtenidas para cada caso de prueba registrar en documento "Reporte de Pruebas VRRPRG6v1.0.docx".

2.1.2. REQUERIMIENTOS NO FUNCIONALES

Lista total de los requerimientos no funcionales que se deben verificar, de acuerdo al plan de verificación de la iteración. Para un detalle mayor de cada iteración ver el Plan de Verificación de la Iteración que corresponda.

Lista de requerimientos no funcionales del componente que se desean verificar de acuerdo al Plan de Verificación de la Iteración:

- Interfaz Gráfica – Atractiva- Amigable para el usuario.
- Facilidad de Uso.
- Funcionabilidad.

Nota: por más detalle de los mismos ver documento de requerimientos.

2.2. WHERE IS MY FRIEND? VERSIÓN 0.0.0.3

Verificadores:

- Dario Britos sobre iPhone
- Gimena Bernardet sobre Windows Phone
- Valentina Da Silva sobre Android
- Victor Diaz sobre Windows 8

2.2.1. REQUERIMIENTOS FUNCIONALES

Lista total de las funcionalidades del componente que se deben verificar, de acuerdo al plan de verificación de la iteración. Para un detalle mayor de cada iteración ver el Plan de Verificación de la Iteración que corresponda.

Lista de funcionalidades del componente que se desean verificar de acuerdo al Plan de Verificación de la Iteración:

- Envío de solicitud.
- Aceptar solicitud.

2.2.1.1. ENVÍO DE SOLICITUD

Entrada	Salida esperada
Seleccionar usuario a seguir. Indicación de tiempo a seguir.	El usuario consigue seguir a un contacto.

Procedimiento de prueba

Iniciar sistema

Dirigirse a la opción de envío de solicitud, realizando cada uno los casos de pruebas indicados por vez. Ver documento: "CP It2 Fase elaboracion Where is my friend.xls".

Acorde a las respuestas obtenidas para cada caso de prueba registrar en documento "Reporte de Pruebas VRRPRG6v1.0.docx".

2.2.1.2. ACEPTAR SOLICITUD

Entrada	Salida esperada
Acepta o Rechaza.	Es seguido o no según entrada.

Procedimiento de prueba

Iniciar sistema

Responder o no la solicitud según corresponda, realizando cada uno los casos de pruebas indicados por vez. Ver documento: "CP It2 Fase elaboracion Where is my friend.xls".

Acorde a las respuestas obtenidas para cada caso de prueba registrar en documento "Reporte de Pruebas VRRPRG6v1.0.docx".

2.2.2. REQUERIMIENTOS NO FUNCIONALES

Lista de requerimientos no funcionales del componente que se desean verificar de acuerdo al Plan de Verificación de la Iteración:

1. Interfaz Gráfica – Atractiva- Amigable para el usuario.
2. Facilidad de Uso.
3. Funcionabilidad.