

Plan de Pruebas - Iteración 2

para

RIDE

PlanPruebas_02

Preparado por:

Ariel Rodríguez Jiménez

Septiembre 25, 2017

Tabla de contenidos

1. Identificador de plan de pruebas	3
2. Referencias	3
3. Introducción	3
4. Ítems de prueba	3
5. Problemas de riesgo de software	4
6. Características a ser probadas	5
7. Características que no se probarán	5
8. Acercamiento	6
8.1 Unidad	6
8.2 Aceptación	6
8.3 Regresión	6
8.4 Indicadores	6
9. Criterios de éxito y fracaso	7
9.1 Criterios de aceptación	7
9.2 Descripción de criterios de fallo	7
9.3 Documentacion de errores	7
10. Criterios de suspensión y requerimientos de reanudación	7
11. Entregables	8
12. Tareas de pruebas pendientes	8
13. Necesidades ambientales	8
14. Necesidades de personal y entrenamiento	9
15. Responsabilidades	9
16. Horario	9
17. Riesgos y contingencias del planeamiento	9
18. Aprobaciones	10
19. Glosario	10

1. Identificador de plan de pruebas

PlanPruebas_02

2. Referencias

El acceso a información adicional para la aplicación 'Ride', se encuentran en distintos documentos desarrollados anteriormente. Pueden ser consultados mediante los siguientes vínculos:

- Documento de Visión :
https://www.dropbox.com/s/esaq0pb9sjer8lx/DocumentoVision_Ariel_Eros_Pablo.pdf?dl=0
- ERS: <https://www.dropbox.com/s/vhxqxt8b12ssge/ERS%20-%20RIDE.pdf?dl=0>
- Primer plan de pruebas:
https://www.dropbox.com/s/cke24hu9le1yuoh/PlanDePruebas_Ride.pdf?dl=0

3. Introducción

En este documento, se realizará una serie de análisis en los siguientes cinco requerimientos de la aplicación seleccionados para entregar en la segunda iteración al cliente, de esta forma, mostrar un avance continuo y estable, asegurando el correcto funcionamiento de la aplicación en sus distintas fases de desarrollo. Se mostrará todo lo que es posible realizarse y también lo que no es posible, principales riesgos que presentan los requerimientos a desarrollar.

Para el desarrollo de este documento, se han utilizado las siguientes herramientas, frameworks y componentes: Angular, Bootstrap, Javascript, Microsoft SQL Server 2008, NodeJS.

4. Ítems de Prueba

Los siguientes puntos corresponden a los requerimientos que se desarrollarán y posteriormente pasarán por una sesión de pruebas para corroborar su correcto funcionamiento.

- Agregar favorito: En caso de que el usuario quiera seguir viajando con cierta persona, la agrega como favorito. De esta manera, cada vez que haya un nuevo viaje se le notifica para que lo tome o no.
- Eliminar favorito: Cuando ya no se quiere tener a un usuario como favorito, se elimina el vínculo y no recibirá más notificaciones de los viajes que realizará el favorito.

5. Problemas de riesgo de software

En el desarrollo de software, siempre existe la posibilidad de que aparezcan distintos problemas, por lo que es importante puntualizar aquellos problemas que pueden ser contemplados con los requerimientos incluidos en este plan de pruebas.

- El software requiere de una constante conexión a internet, de no ser así, ocurrirá el caso en el que los usuarios no podrán agregar favoritos ni quitar favoritos
- Dependiendo del tipo de usuario que haga uso de estos requerimientos, es posible que existan usuarios a los que no les guste el aspecto visual de las distintas páginas de la aplicación o no entiendan cómo realizar correctamente las acciones que desea.
- Existe la posibilidad de que al momento de la programación, surjan nuevas características que agregar a los requerimientos ya estipulados, esto implicaría un aumento en el tiempo de trabajo.
- Se debe tener en cuenta de que en cualquier momento, un desarrollador del equipo se puede discapacitar, por lo que el tiempo de desarrollo podría verse afectado.

6. Características a ser probadas

Se pondrán a prueba los siguientes puntos para las funcionalidades previamente mencionadas en este documento. Estos se calificarán en cuanto a facilidad de comprensión y entendimiento de uso en Alto (A), Medio (M) y Bajo (B).

-
- Facilidad para agregar como favorito a un usuario. (A)
- Facilidad para eliminar como favorito a un usuario. (A)
- Correcto funcionamiento para agregar usuario a favoritos. (M)
- Correcto funcionamiento para eliminar usuario de favoritos. (M)

7. Características que no se probarán

Los siguientes puntos corresponden a características generales relacionales a los requerimientos a desarrollar en este documento, que no se analizarán, desarrollarán o probarán para la entrega de la segunda iteración.

- Errores de hardware por parte de los usuarios, que impidan la ejecución del sistema.

- Posibilidad de errores con la conexión a internet, que impidan el movimiento de datos.
- Todos los errores especificados en el documento de especificación de requerimientos.

8. Acercamiento

8.1 Unidad

Los siguientes puntos corresponde al análisis individual que se le dará a los requerimientos a desarrollar en la segunda iteración.

- Agregar como favorito a un usuario por parte de otro.
- Eliminar de favoritos a un usuario por parte de otro.

8.2 Aceptación

Los siguientes puntos corresponden a los puntos que deben estar completos y funcionales para la aceptación completa de la segunda iteración de requerimientos.

- Agregar usuario como favorito
- Eliminar usuario como favorito

8.3 Regresión

Para esta segunda iteración, si ocurriera algún error en el sistema con una funcionalidad en específico y se cumplen con las características requeridas para asegurar su correcto funcionamiento, el sistema retornará al usuario a la página de inicio para así realizar nuevamente el proceso que tuvo errores.

8.4 Indicadores

- El equipo de trabajo estará formado por tres desarrolladores que se distribuirán los requerimientos a entregar en esta segunda iteración de forma equitativa. La entrega, consultas o errores serán responsabilidades del equipo y no asuntos individuales.
- Se pondrán a prueba solamente las versiones finales programadas de cada requerimiento.
- Actualmente no se tiene planeado ningún mantenimiento o extensión de estos requerimientos para el futuro.

9. Criterios de éxito y fracaso

9.1 Criterios de aceptación

- Funcionalidad de los requerimientos en un 97%. Esto debido a que en cuanto a software, es imposible asegurar un 100%, por lo que se deja un 3% de anuencia a distintos errores mencionados en la sección de problemas de riesgo.
- Cumplimiento de desarrollo de todos los requerimientos estipulados.
- Cumplimiento de las etapas de diseño, desarrollo, análisis de pruebas e implementación.

9.2 Descripción de criterios de fallo

Dado que es imposible asegurar un 100% de funcionamiento de la aplicación, y para 'Ride' se asegura un 97%, existe la posibilidad de que para el uso futuro del sistema, en algún momento se pueda presentar un fallo. Se considera fallo cuando el resultado de una funcionalidad no ha sido el esperado, se asegura en un 97% la corrección de los fallos que puedan ocurrir, pero de igual manera es imposible abarcar el 100% de estos durante la etapa de pruebas.

9.3 Documentación de errores

Se mantendrá un registro e historial de errores que se irá actualizando en caso de que aparezca algún error en los requerimientos durante la etapa de pruebas, cada registro incluirá su plan de solución para posteriormente realizarlo.

10. Criterio de suspensión y requerimientos de reanudación

- Será motivo de suspensión el proceso de pruebas si no se tienen desarrollados como mínimo cuatro de los requerimientos estipulados.
- La reanudación del proceso de desarrollo de algún requerimiento será hasta que los errores encontrados para el mismo se hayan resuelto exitosamente.

11. Entregables

Los entregables hasta la fecha para la segunda iteración serán los siguientes:

- Minuta sobre reuniones de los desarrolladores con los clientes.
- Documento de visión y alcance del sistema.
- Documento de especificación de requerimientos.
- Primer plan de pruebas para primera iteración:

- Registro/Inicio de sesión de un usuario
- Visualización de perfil de usuario
- Registrar vehículo
- Modificar vehículo
- Eliminar vehículo
- Segundo plan de pruebas para segunda iteración:
 - Agregar favorito
 - Eliminar favorito
 - Bloquear usuario
 - Desbloquear usuario
 - Crear viaje

12. Tareas de pruebas pendientes

Las siguientes son los requerimientos pendientes por realizar para las futuras iteraciones.

- Modificar Viaje
- Eliminar Viaje
- Aceptar Pasajero
- Buscar Viaje
- Ver Viaje
- Reservar Viaje
- Cancelar Reservación

Cada uno de estos requerimientos incluyen su plan de pruebas y registro de errores.

13. Necesidades ambientales

Para el desarrollo de la aplicación y de las pruebas necesarias para cada requerimiento, se necesita de una computadora con los requerimientos técnicos necesarios para soportar las herramientas de NodeJS y SQL Server 2008, así como la capacidad de simular/virtualizar un dispositivo móvil para realizar las pruebas de los requerimientos una vez desarrollados.

14. Necesidades de personal y entrenamiento

Para el correcto desarrollo de la aplicación deseada, el equipo de desarrollo debe tener conocimientos en Javascript, Angular, Microsoft SQL Server, NodeJS, HTML, CSS, Bootstrap. Ante el desconocimiento de alguna de estas tecnologías, se

considerará como tiempo efectivo de trabajo, el tiempo utilizado por cada desarrollador para familiarizarse con las tecnologías.

15. Responsabilidades.

Los tres desarrolladores trabajarán de forma equitativa. El coordinador del equipo debe encargarse y asegurar el correcto avance y funcionamiento de los distintos requerimientos durante el proceso de desarrollo y de pruebas, de igual manera el cumplimiento de las características de cada uno. El equipo de desarrollo se hace responsable de las etapas de desarrollo, fases y errores de software que surjan durante el tiempo establecido del proyecto.

16. Horario.

Se trabajará en promedio 10 horas semanales por cada miembro del equipo. Se deberá destinar un mínimo de 2 horas para la etapa de pruebas de cada caso de uso. En caso de presentar errores durante el desarrollo, se debe destinar un tiempo acorde a la correcta detección del error e implementación de la solución. Una vez finalizadas las etapas de diseño, desarrollo, pruebas e implementación, se espera que el tiempo destinado para cambios o mantenimientos sea mínimo y no interrumpa el proceso de desarrollo de los próximos requerimientos.

17. Riesgos y contingencias del planeamiento.

Los siguientes puntos representan los riesgos generales de la etapa de desarrollo junto con sus planes de contingencia:

- Existe la posibilidad de atrasos al momento del desarrollo programado de los requerimientos, puede ser por falta de conocimiento sobre el uso de alguna tecnología o la ausencia de algún miembro del equipo. Dado cualquiera de estos casos, el tiempo destinado al desarrollo del proyecto necesitará aumentar en al menos 3 horas.
- Existe la posibilidad de que el cliente no quede completamente satisfecho dados los resultados obtenidos por la aplicación, aún cuando todos los requerimientos se han cumplido como fueron establecidos, dado este caso se procederá a realizar cambios en los requerimientos para ajustarse a los nuevos gustos del cliente, mientras que estos cambios no superen las 5 horas además de las ya destinadas para los requerimientos aún por desarrollar.
- En caso de que en alguna iteración no se cumpla con los requerimientos propuestos a desarrollar, estos deberán ser finalizados para la siguiente

iteración, sin disminuir la prioridad ni el tiempo destinado de trabajo para los otros requerimientos.

18. Aprobaciones.

Al finalizar el proyecto 'Ride', esta contará con la aprobación de la Escuela de Computación del Instituto Tecnológico de Costa Rica para poder utilizar su sello y nombre en los documentos y en el sistema como tal. Se contará con la aprobación de DATIC, por lo que la base de datos del sistema se tendrá en sus servidores. Una vez aprobado el sistema por parte de los clientes, la aplicación será entregada.

19. Glosario.

Angular: Infraestructura basada en Javascript.

Bootstrap: Framework para desarrollo y diseño de aplicaciones web.

CSS: Lenguaje de diseño gráfico.

DATIC: Departamento de Administración de Tecnologías de Información y Comunicaciones.

Framework: Ambiente de trabajo.

HTML: Elemento de construcción básico para una página web.

Javascript: Lenguaje orientado a objetos.

Ride: Nombre de la aplicación en desarrollo, su significado traducido al español es 'Viaje'.

SQL Server: Sistema administrador de bases de datos relacionales.

NodeJS: Librería de JavaScript y entorno de ejecución de E/S.

- Final del Documento -