

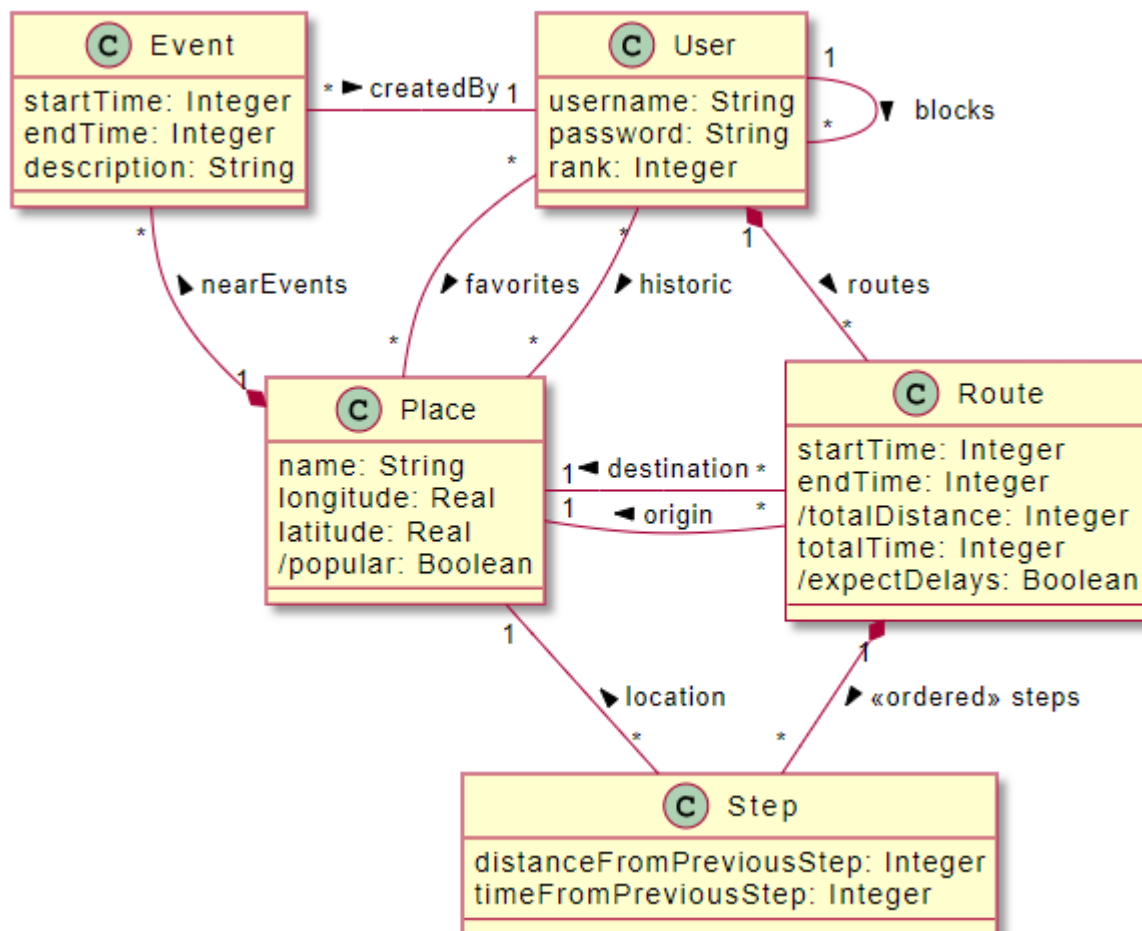
Ingeniería de Requisitos

PEC 4: Documentación en OCL y V & V de Requisitos

Pregunta 1 (40%)

Enunciado

Supongamos que estamos trabajando en el análisis del modelo de datos de la aplicación colaborativa de navegación y tránsito y disponemos, por el momento, del siguiente diagrama de clases UML:



Notas: Los atributos *startTime* y *endTime* son enteros que representan un timestamp en [formato estándar de Unix](#). Los atributos *totalDistance* y *distanceFromPreviousStep* están expresados en metros y *totalTime* y *timeFromPreviousStep* están expresados en segundos.

Escribe en OCL las restricciones que se piden a continuación:

- a) [10%] Un lugar es popular si ha sido marcado como favorito por 2 o más de 2 usuarios.

Solución:

context Place::popular:Boolean **derive**:

self.userFavorited->size() >= 2

- b) [12,5%] El destino de una ruta (*destination*) tiene que ser igual a la localización (*location*) del último paso de esa ruta.

Solución:

context Route **inv**:

self.destination = self.step->last().location

- c) [12,5%] La distancia total de una ruta (*totalDistance*) es el resultado de sumar la distancia (*distanceFromPreviousStep*) de todos los pasos de esa ruta.

Solución:

context Route::totalDistance:Integer **derive**:

self.step.distanceFromPreviousStep->sum()

- d) [5%] En una ruta esperaremos retrasos (*expectDelays*) cuando existan eventos (*nearEvents*) en alguno de los lugares por los que pasa. Recuerda que puedes consultar los lugares por los que pasa una ruta a partir de su origen (*origin*) y sus pasos (*location* en *Step*).

Solución:

context Route::expectDelays:Boolean **derive**:

self.origin.nearEvents->size(>0) or self.step->exists(s|s.location.nearEvents->size(>0))

Pregunta 2 (30%)

Enunciado

Supongamos que nos han pedido especificar el sistema descrito en la pregunta anterior y que hemos identificado los siguientes requisitos:

[REQ-1] El sistema no mostrará ni los mensajes ni los eventos de los usuarios que previamente hayan sido bloqueados.

[REQ-2] El sistema almacenará los lugares favoritos de cada usuario así como sus últimos destinos.

[REQ-3] El sistema siempre proporcionará la mejor ruta para el destino solicitado.

[REQ-4] El rango del usuario se basará en el número de eventos que comunique.

[REQ-5] Para cada paso de ruta el sistema mostrará los eventos cercanos para que el conductor pueda actuar en consecuencia.

Indica 3 problemas que detectes con este conjunto de requisitos indicando, para cada problema, cuál de las propiedades deseables de una especificación (que sea correcta, no ambigua, completa, consistente, ordenada, verificable, modificable y trazable) es la que no cumple debido a aquel problema.

A continuació tienes un ejemplo:

Propiedad: Completa

Problema detectado: La especificación no es completa porque faltan muchos requisitos

Solución

Propiedad: No ambigua

Problema detectado: REQ-4 no indica exactamente cómo calcular el rango a partir del número de eventos que comunique el usuario.

Propiedad: Verificable

Problema detectado: REQ-3 no se puede verificar ya que "mejor" es una apreciación subjetiva y no definida: ¿más rápida en tiempo? ¿con menor distancia? ¿con menor tráfico?

Propiedad: Consistente

Problema detectado: REQ-1 y REQ-5 no son compatibles ya que en REQ-1 se habla de bloquear mensajes y eventos de un usuario, y en el REQ-5 se dice que se mostrarán los eventos cercanos, ¿que ocurre con los eventos cercanos de un usuario bloqueado?

Pregunta 3 (30%)

Waze es una plataforma como la que hemos estado especificando, con múltiples funcionalidades colaborativas para la navegación y el tránsito. En concreto, nos centraremos en su buscador disponible en <https://www.waze.com/>. También lo encontraremos disponible para plataformas Android e iOS.

Supón que nos han proporcionado la siguiente especificación del buscador para verificarlo:

- El buscador propone resultados a medida que el usuario escribe en la caja de texto:
 - Detecta automáticamente todas las localidades y lugares cuyo nombre coincide con el fragmento introducido por el usuario (por ejemplo, si introducimos “cuenc” sugiere, entre otros, “Cuenca”).
 - Detecta destinos sin tener en cuenta las mayúsculas y minúsculas (por ejemplo, si introducimos “cuENca” sugiere “Cuenca”).
 - Detecta destinos sin tener en cuenta los acentos introducidos por error (por ejemplo, si introducimos “cuéncà” sugiere “Cuenca”).
 - Detecta destinos sin tener en cuenta caracteres especiales (por ejemplo, si introducimos “c’u’en’ca” sugiere “Cuenca”).
- a) Haz un plan de pruebas de conformidad que nos permita ver si el buscador de Waze se comporta tal y como se ha especificado. Es importante que el plan de pruebas tenga el contenido que se indica en el apartado 2.2 del módulo 5 de los materiales.
- b) Ejecuta el plan de pruebas y documenta también el resultado del mismo de acuerdo a los entregables que hayas indicado en el plan de pruebas.

Solución

Plan de pruebas de conformidad

Identificador:

- PC01

Referencias:

- Enunciado de la PEC4

Elementos de prueba:

- Buscador de la plataforma Waze

Riesgos identificados:

- Ninguno

Características que no se probarán

- Ninguno

Estrategia de pruebas

- Abriremos la web <https://www.waze.com/es/live-map/> en un navegador e introduciremos un texto en el buscador para ver qué palabras ha sugerido
- Centraremos el mapa en nuestra posición actual para que el sistema nos devuelva resultados cercanos, sino obtendremos resultados de Estados Unidos.
- Para recuperar el resultado haremos una captura de pantalla del mismo
- Ejecutaremos un ejemplo de cada requisito y recogeremos el resultado

Pruebas a realizar

E1. Fragmentos:

Texto a introducir: "cuenc"

Resultado esperado: "Cuenca"

E2. Mayúsculas:

Texto a introducir: "cuENca"

Resultado esperado: "Cuenca"

E3. Acentos:

Texto a introducir: "cuéncà"

Resultado esperado: "Cuenca"

E4. Caracteres especiales:

Texto a introducir: "c'u'en'ca"

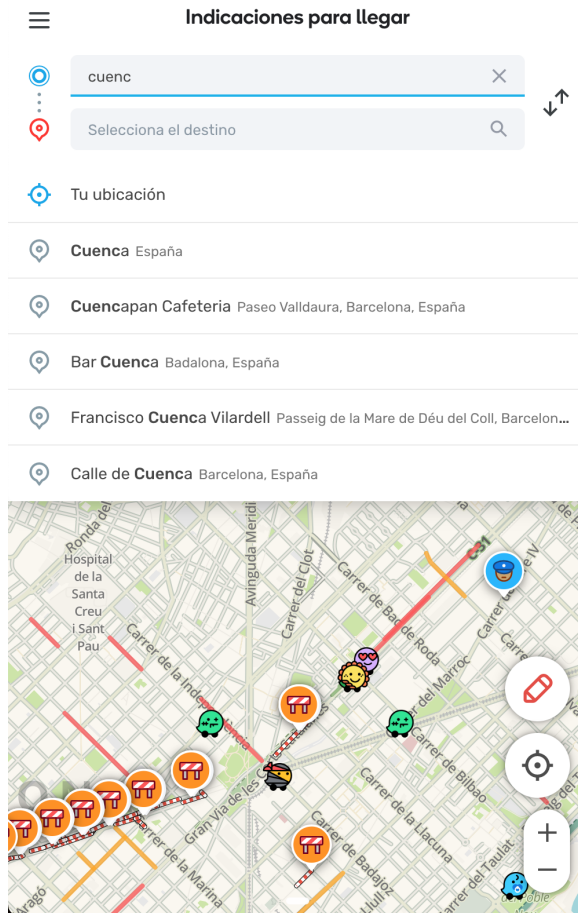
Resultado esperado: "Cuenca"

Entregables

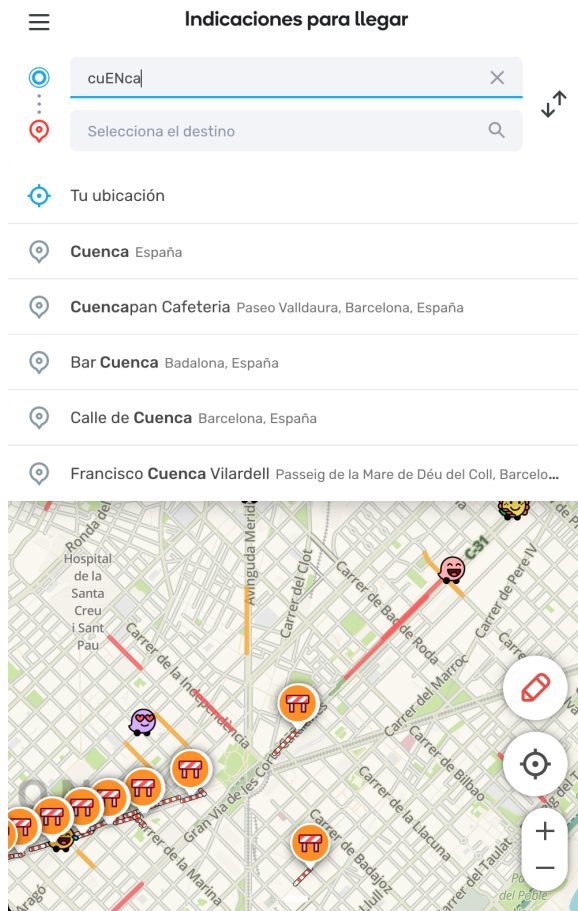
- El plan de pruebas
- Para cada caso de prueba, captura de pantalla con el resultado obtenido así como el resultado de la prueba (correcto o error).

Resultado de la ejecución:

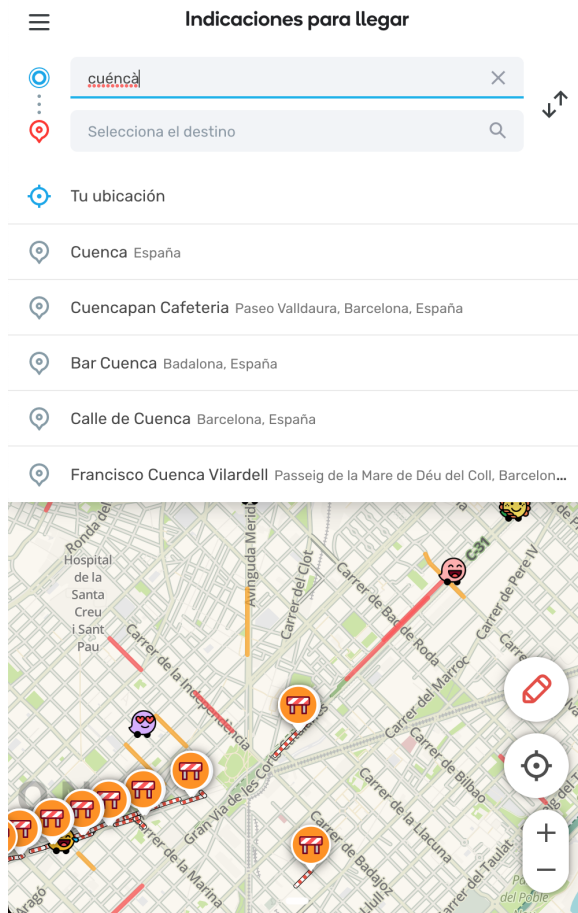
E1. Fragmentos (correcto)



E2. Mayúsculas (correcto)



E3. Acentos (correcto)



E4. Caracteres especiales (correcto)

