

# Especificação de Software

## Relatório do primeiro projecto

Jorge Veiga (75154)

André Santos (84912)

Rodrigo Bernardo (78942)

### Estratégia de Refinamento

A nossa estratégia é uma estratégia de refinamento horizontal. Esta passa por, gradualmente, formalizar as definições e os requisitos pela ordem que aparecem no enunciado.

### Máquinas

1. *zober0*: Introduz os conceitos referentes aos clientes (secção 2.1). Implementa as operações *newClient*, *removeClient*, *upgradePlan* e *downgradePlan*, e foca-se nas restrições 1 a 10 (secção 2.6). A parte estática está modelada no contexto *c0* (conjuntos *CLIENTS*, *NAMES*, *EMAILS* e *PLAN*, e constantes *REGULAR* e *VIP*).
2. *zober1*: Refina a máquina *zober0* e introduz os conceitos referentes aos condutores (secção 2.2.). Implementa as operações *newDriver*, *removeDriver* e *banDriver* e foca-se nas restrições 11 a 17. A parte estática está modelada em *c1*, contexto este que estende *c0* com os conjuntos *LICENSES* e *DRIVERS*.
3. *zober2*: Refina a máquina *zober1* e introduz os conceitos referentes aos carros (secção 2.3). Implementa as operações *addCar*, *removeCar*, *addDriverToCar*, *removeDriverFromCar*, *upgradeService* e *downgradeService*. As restrições focadas são da 18 até à 28. A parte estática está modelada em *c2*, contexto este que estende *c1* com os conjuntos *CARS* e *ZOBERSERVICE*, e com as constantes *ZoberY* e *ZoberWhite*.
4. *zober3*: Refina a máquina *zober2* e introduz os conceitos referentes às viagens (secção 2.4). Implementa as operações que faltam, *newRide*, *cancelRide* e *completeRide*, e foca-se nas restrições que 29 a 40, não faltando nenhuma por modelar. A parte estática encontra-se no contexto *c3*, que estende *c2* com o conjunto *RIDE*.
5. *zober\_sharing\_m0*: Introduz os conceitos relativos ao *ZoberSharing*. Implementa uma nova operação, *newSharedRide*, que permite programar uma nova viagem para o conjunto de clientes passado como parâmetro no intervalo especificado para um carro que esteja disponível com o serviço pretendido. É uma operação análoga a *newRide*, mas garante as resrtições enunciadas na secção 4. Na nossa solução, não procuramos atribuir os melhores carros. Introduz uma única variável, *shared\_rides*, que é uma relação entre viagens ainda não completadas, tal que se o par  $(x,y)$  lhe pertence, então *x* está a ser partilhada com *y*. Com a introdução desta operação tivemos que modificar de acordo as operações já existentes *banDriver*, *completeRide* e *cancelRide*. Em particular, na nossa solução uma viagem pode ser cancelada não cancelando aquelas às quais está ligada. De forma análoga, cada viagem deve ser completada individualmente.