Changelog

# Changes in UML domain model

#### Título de ejemplo

* Lo dejo para saber el formato.

# Changes in JAVA domain model

# Changes in populateDatabase.xml



# Changes in Repositories

#### CreditCardRepository

* + The methods findSponsorshipByCreditCard(id) and findApplicationByCreditCard(id) are moved from CreditCardRepository to SponsorshipRepository and ApplicationRepository respectively. Because both methods return a collection of applications and a collection of sponsorships instead of a Credit card or collection of credit cards.

#### FinderRepository

* + The method findFixUpTaskFinder(keyWord, startPrice, endPrice, startDate, endDate, warranty, category, pageable) is moved from FinderRepository to FixUpTaskRepository because the method returns a collection of fix-up tasks instead of a Finder or collection of Finders.

#### SectionRepository

* + The methods findHandyWorkerBySection(id) and findTutorialBySection(id) are moved from SectionRepository to HandyWorkerRepository and TutorialSection respectively because both methods don’t return Section objets.

# Changes in Services:

#### ActorService

* + The methods isBanner(actor) and notBanner(actor) has been replaced by changeBanner(actor). ChangeBanner(actor) integrate the functionality of isBanner(actor) and notBanner(actor) in a unique procedure.

#### ApplicationService

* + The method checkByPrincipal(application) has been created. It helps to refactoring. It´s invoked twice: ApplicationService::save and ApplicationService::delete.
  + The methods removeApplicationToHandyWorker(application) and removeApplicationToFixUpTask(application) have been simplified. Those methods contained unnecessary code.

#### boxService

* + boxService::create has been refactoring so that method is more readable.
  + The method checkByPrincipal(box) has been created. It helps to refactoring. It´s invoked twice: boxService::save and boxService::delete.
  + The private method has been created: this.checkName(box). It checks that not exists a custom box whose name is “in box”, “out box”, “trash box” and “spam box”. This method is invoked by boxService::save.
  + The method createDefaultBox(actor) has been refactorized. Now, it’s more readable.
  + The method createDefaultBox(actor) is now a procedure.

#### CurriculumService

* + Added the method findByPrincipal(). This method returns the curriculum Principal’s. It helps to refactoring.

#### EndorsementService

* + Now, EndorsementService::playedRole is a private method by security reasons.

#### FinderService

* + The method checkByPrincipal(finder) has been created. It helps to refactoring. It´s invoked twice: FinderService::save and FinderService::search.

#### FixUpTaskService

* + Added the method checkByPrincipal(fixUpTask): check that the principal can edit or delete the object. It helps to refactoring.

#### MessageService

* + boxService::create has been refactoring so that method is more readable.
  + The method messageToStatus(application, status) has been simplified.

#### SectionService

* + Added the method checkByPrincipal(section): check that the principal can edit or delete the object. It helps to refactoring.

#### SocialProfileService

* + Added the method checkByPrincipal(socialProfile): check that the principal can edit or delete the object. It helps to refactoring.

#### SponsorshipService

* + Added the method checkByPrincipal(sponsorship): check that the principal can edit or delete the object. It helps to refactoring.

#### TutorialService

* + Added the method checkByPrincipal(tutorial): check that the principal can edit or delete the object. It helps to refactoring.

#### PersonalRecordService, EducationallRecordService, ProfessionalRecordService, MiscellaneousRecordService and EndorserRecordService

* + Added checkByPrincipal(TRecord): this procedure check if the principal can edit or delete the record “TRecord” received as argument.
  + The methods save() and deleted() invoke checkByPrincipal().

#### UtilityService

* + The method checkDate(start, end) has been created: check that start moment must be before than end moment. This methos is invoked in different services several times: EducationRecordService::save, ProfessionalRecordService::save, PhaseService::save, FixUpTaskService::save, NoteService::save and ReportService::save.
  + The method current\_moment() has been created. It helps to refactoring. It is invoked in several services: ApplicationService,::save TutorialService::save, FinderService::save, FixUpTaskService::save, ComplaintService::save, ReportService::save, NoteService::save, EndorsementService::save and MessageService::save.