



Documento de Cambios

Grupo: 20

Miembros:

José Ángel Domínguez Espinaco

Daniel Lozano Portillo

José Joaquín Rodríguez Pérez

María Ruiz Gutiérrez

Miguel Ternero Algarín

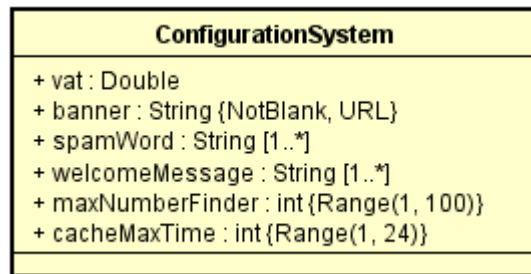
Laura Vera Recacha

Contenido

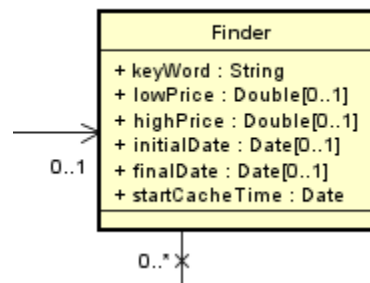
Cambios en UML:	3
Cambios en JAVA:.....	6
Cambios en REPOSITARIOS:	7
Cambios en SERVICIOS:	8

Cambios en UML:

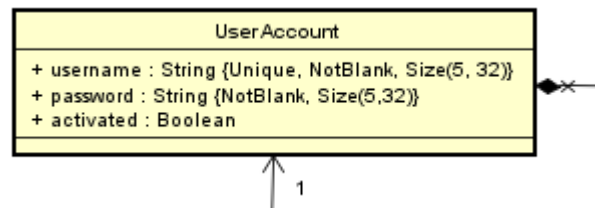
- Añadido atributo `maxNumberFinder` y `cacheMaxTime` en la clase `ConfigurationSystem`. El primero indica el número máximo de objetos que puede devolver el buscador y el segundo indica el tiempo máximo en horas que debe guardarse en caché una búsqueda realizada por un `explorer`.



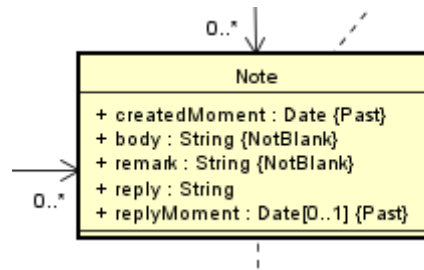
- Añadida clase `Finder` y relacionada con `Explorer` y `Trip`. Esa clase indica la búsqueda realizada por un `Explorer` en concreto y se guarda en la base de datos.



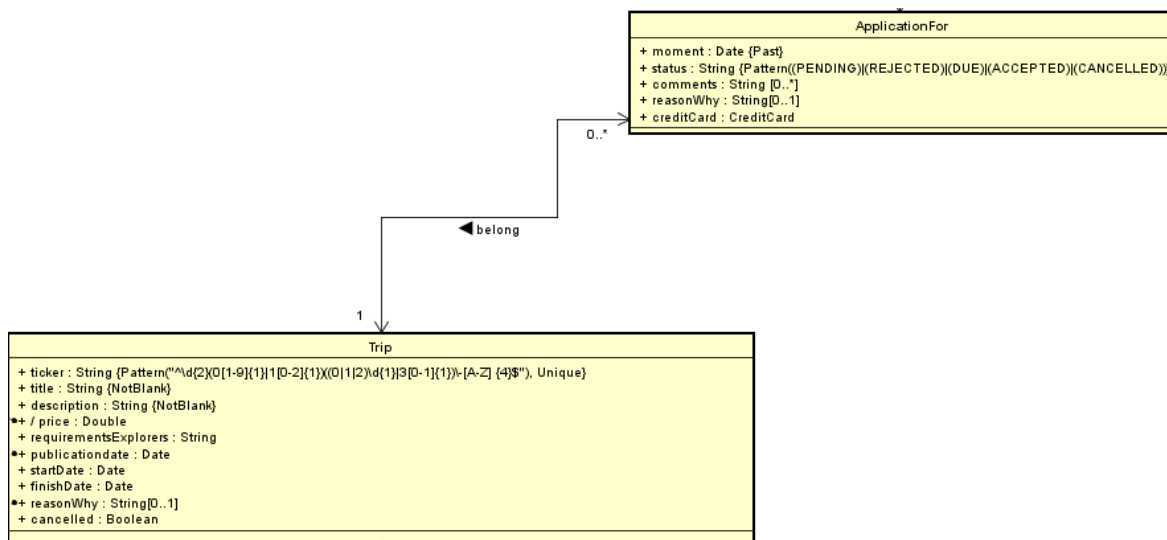
- Introducido atributo `activated` en `UserAccount` para saber si una cuenta esta activada o no.



- El atributo remark de la clase Note ha sido cambiado a String, también se ha modificado en esta misma clase el atributo reply. Se ha puesto como notnull.



- Cambio navegabilidad trip y applicationfor, pasa a ser bidireccional para poder implementar la parte de manager con applications.



Cambios en JAVA:

Nota: En los siguientes cambios no se adjuntarán imágenes para no hacer demasiado pesada la lectura del documento con las capturas del código. Para ver el código diríjase al proyecto a la clase y al método especificado en cada cambio.

- Añadido atributo `maxNumberFinder` y `cacheMaxTime` en la clase `ConfigurationSystem`.
- Añadida una nueva clase llamada `Finder` que indica la búsqueda realizada por un `Explorer` en concreto.
- Añadido atributo `Finder` en la clase `Explorer` indicando la relacion one to one.
- Debido a la introducción del atributo `activated` se ha modificado el metodo `isEnabled` de `userAccount`.
- El atributo `remark` de la clase `Note` ha sido cambiado a `String`.

Cambios en REPOSITARIOS:

- En la clase TripRepository se ha cambiado la query de findAllTripsNoAuthenticate(), ahora se muestran todas las trips guardados en la BD.
- En la clase TripRepository se ha añadido la query findAllTripsByKeyWord(String search, Pageable pageable). El pageable es para que se devuelvan solamente un número determinados de objetos (los que indica la propiedad maxNumberFinder de la clase ConfigurationSystem)
- Se ha añadido el repositorio relacionado con la clase Finder.
- En la clase CurriculaRepository se han añadido los métodos siguientes para evitar que un Ranger edite el curriculum de otro Ranger:
 - Curricula CurriculaWithThisPersonalRecord(int personalRecordId);
 - Curricula CurriculaWithThisMiscellaneousRecord(int miscellaneousRecordId);
 - Curricula CurriculaWithThisEndorserRecord(int endorserRecordId);
 - Curricula CurriculaWithThisProfessionalRecord(int professionalRecordId);
 - Curricula CurriculaWithThisEducationRecord(int educationRecordId);

Cambios en SERVICIOS:

- Se han implementado los servicios de los métodos de los repositorios comentados anteriormente.
- Se ha añadido en todos los create de todos los actores la siguiente línea de código:
`userAccount.setActivated(true);`
- Se ha añadido en todos los create de todos los actores la siguiente línea de código:
`messagesFolders.addAll(this.messageFolderService.createDefaultFoldersForRegister(result));`

Debido a que el anterior método `createDefaultFolders()` solo creaba las carpetas si el actor estaba autenticado y aún no puede estar autenticado porque se está registrando en el sistema.

- Se ha cambiado el método `save` de todos los Actores añadiendo que si el actor se está registrando en el sistema por primera vez, la contraseña que se almacene en la BD sea el Hash y no la contraseña que él ha puesto en el input del formulario.
- Se ha modificado el método `save` y `delete` de `MiscellaneousRecordService`, `PersonalRecordService`, `EndorserRecordService`, `EducationRecordService`, `ProfessionalRecordService` de manera que cuando se guarda o se elimina uno de éstos se actualiza los cambios en la clase `Curricula` a la que está unida ya que la relación es unidireccional.
- Cambio en el Servicio de message para que cuando se elimine un message vaya a la carpeta trash y si se elimina de dicha papelera se borre de la Base de Datos.
- Añadido el método para `sendMessageToActorOfApplicationFor` para que cuando se cambie el status de una solicitud envíe un message a los actores participantes en la solicitud.
- Añadido el método `sendNotificationBroadcas` que permite a un administrador mandar un mensaje a varios actores del sistema.
- El atributo `reply` de `Note` se ha comprobado con un metodo que no contenga espacios en blanco, lo cual no se podrá poner una reply con espacios.
- Cambio en el método `generatedTicker` en el servicio de Trip ya que para los días con un solo dígito no lo teníamos en cuenta y no le generaba el 0 delante.
- Se ha modificado por completo el método `delete` de trip ya que al quitar bidireccionalidades tenemos que tratar las unidireccionalidades que tuvieran FK asociadas

a la trip que queramos eliminar, eliminando antes su asociación y por último la trip.