

Verificación de software

Verificación de un sistema bla bla

Autor:

Juan Ignacio Farizano

Trabajo Práctico Final - Ingeniería de Software II

Departamento de Ciencias de la Computación

Facultad de Ciencias Exactas, Ingeniería y Agrimensura

Universidad Nacional de Rosario

Rosario, Santa Fe, Argentina

30 de enero de 2025

El problema

Queremos especificar el funcionamiento del sistema *Mis Estudiantes* que utilizan directivos de escuelas primarias y secundarias para inscribir y reinscribir alumnos (por ej. promoción de grado, repitente, etc) en la base de datos del Ministerio de Educación de la provincia de Buenos Aires, lo descrito es similar al sistema existente pero simplificado para mantener una dificultad razonable para este trabajo práctico.

Un directivo desea inscribir o reinscribir a un alumno en su escuela. Cada alumno cuenta con un legajo donde cada entrada una representa un paso del alumno por el sistema educativo; cada registro describe el grado en que el alumno fue inscripto y tiene tres estados posibles:

- **Inscripto:** el alumno fue inscripto y es habilitado a cursar el grado registrado
- **Promueve:** el alumno cumplió los requisitos para promocionar el grado al cual se encuentra inscripto y es habilitado a ser inscripto en el grado siguiente o a graduarse si se encontraba en el 12avo grado
- **Repite:** el alumno no cumplió los requisitos para promocionar de grado y debe ser inscripto en el mismo grado

Además se desea llevar un registro fácilmente accesible de los alumnos graduados, es decir alumnos que haya promocionado 12avo grado.

Se deben especificar las siguientes operaciones:

1. **Inscribir:** un directivo inscribe por primera vez a un alumno a primer grado o a un alumno pre-existente al grado que corresponda según su último estado de inscripción
2. **Cerrar inscripción:** la inscripción actual es cerrada correspondientemente a si cumplió o no los requisitos para promocionar de grado
3. **Consultar repitencia:** se desea consultar si un alumno repitió el último grado que haya cursado

Se asume que se registrará el historial completo del alumno, empezando siempre desde primer grado hasta que finalice. Para simplificar no diferenciamos entre primaria y secundaria, registramos desde 1er grado hasta 12avo, el último del secundario. Los requisitos para promocionar de grado se encuentran por fuera del sistema y no deben ser tenidos en cuenta. Se quieren saber todos los estados que pasaron las inscripciones, por lo tanto cada estado de cada inscripción es inmutable, si un alumno repite se produce un nuevo registro con estado **Inscripto** en el mismo grado y si promociona se registra nuevamente con el estado **Promueve**.

Nota: Esto quizás es demasiado para el enunciado, mucho puede moverse a los comentarios entre los

esquemas más abajo

Designaciones

Especificación en Z

$[ALUMNO]$

$GRADO == \mathbb{N}$

$ESTADO ::= inscripto \mid promueve \mid repite$

$REP ::= alumnoEsRepitente \mid alumnoNoEsRepitente \mid alumnoNoEncontrado$

MisEstudiantes

$legajos : ALUMNO \rightarrow \text{seq } GRADO \times ESTADO$

$graduados : \mathbb{P} ALUMNO$

RequisitoGraduadoInv

MisEstudiantes

$\forall a \in graduados \bullet (last(legajos\ a)).1 = 12 \wedge (last(legajos\ a)).2 = promueve$

GraduadoTieneLegajoInv

MisEstudiantes

$graduados \subseteq \text{dom } legajos$

MisEstudiantesInicial

MisEstudiantes

$legajos = \emptyset$

$graduados = \emptyset$

InscribirAlumnoNuevoOk

$\Delta MisEstudiantes$

$alumno? : ALUMNO$

$alumno? \notin \text{dom } legajos$

$legajos' = legajos \cup \{alumno? \mapsto \langle (1, inscripto) \rangle\}$

$graduados' = graduados$

InscribirAlumnoPromovidoOk

$\Delta MisEstudiantes$

$alumno? : ALUMNO$

$alumno? \in \text{dom } legajos$

$(last (legajos alumno?)).2 = promueve$

$1 \leq (last (legajos alumno?)).1 < 12$

$legajos' = legajos \oplus \{alumno? \mapsto legajos alumno? \frown \langle ((last (legajos alumno?)).1 + 1, inscripto) \rangle\}$

$graduados' = graduados$

InscribirAlumnoRepitenteOk

$\Delta MisEstudiantes$

$alumno? : ALUMNO$

$alumno? \in \text{dom } legajos$

$(last (legajos alumno?)).2 = repite$

$1 \leq (last (legajos alumno?)).1 \leq 12$

$legajos' = legajos \oplus \{alumno? \mapsto legajos alumno? \frown \langle ((last (legajos alumno?)).1, inscripto) \rangle\}$

$graduados' = graduados$

InscribirAlumnoGraduadoE

Ξ *MisEstudiantes*

alumno? : *ALUMNO*

alumno? \in dom *legajos*

$(last (legajos\ alumno?)).2 = promueve$

$(last (legajos\ alumno?)).1 = 12$

InscribirAlumnoDobleInscripE

Ξ *MisEstudiantes*

alumno? : *ALUMNO*

alumno? \in dom *legajos*

$(last (legajos\ alumno?)).2 = inscripto$

$InscribirAlumnoE == InscribirAlumnoGraduadoE \vee InscribirAlumnoDobleInscripE$

$InscribirAlumnoOk == InscribirAlumnoNuevoOk \vee InscribirAlumnoPromovidoOk$

$\vee InscribirAlumnoRepitenteOk$

$InscribirAlumno == InscribirAlumnoOk \vee InscribirAlumnoE$

CerrarInscripcionNoGraduadoOk

Δ *MisEstudiantes*

alumno? : *ALUMNO*

estado? : *ESTADO*

alumno? \in dom *legajos*

$estado? = repite \vee (estado? = promueve \wedge (last (legajos\ alumno?)).1 < 12)$

$(last (legajos\ alumno?)).2 = inscripto$

$legajos' = legajos \oplus \{alumno? \mapsto legajos\ alumno? \frown \langle ((last (legajos\ alumno?)).1, estado?) \rangle\}$

CerrarInscripcionGraduadoOk

Δ MisEstudiantes

alumno? : ALUMNO

estado? : ESTADO

alumno? \in dom *legajos*

estado? = *promueve*

(last (*legajos alumno?*)).1 = 12

(last (*legajos alumno?*)).2 = *inscripto*

legajos' = *legajos* \oplus {*alumno?* \mapsto *legajos alumno?* \wedge \langle ((last (*legajos alumno?*)).1, *estado?*)) \rangle }

graduados' = *graduados* \cup {*alumno?*}

CerrarInscripcionEstadoInvalidoE

Ξ MisEstudiantes

estado? : ESTADO

estado? = *inscripto*

CerrarInscripcionAlumnoNoEncontradoE

Ξ MisEstudiantes

alumno? : ALUMNO

alumno? \notin dom *legajos*

$CerrarInscripcionE == CerrarInscripcionEstadoInvalidoE \vee CerrarInscripcionAlumnoNoEncontradoE$

$CerrarInscripcionOk == CerrarInscripcionNoGraduadoOk \vee CerrarInscripcionGraduadoOk$

$CerrarInscripcion == CerrarInscripcionOk \vee CerrarInscripcionE$

AlumnoEsRepitenteSiOk

$\Xi \text{MisEstudiantes}$

$\text{rep!} : \text{REP}$

$\text{alumno?} \in \text{dom legajos}$

$(\text{last}(\text{legajos alumno?})).2 = \text{repite}$

$\text{rep!} = \text{alumnoEsRepitente}$

AlumnoEsRepitenteNoOk

$\Xi \text{MisEstudiantes}$

$\text{rep!} : \text{REP}$

$\text{alumno?} \in \text{dom legajos}$

$(\text{last}(\text{legajos alumno?})).2 \neq \text{repite}$

$\text{rep!} = \text{alumnoNoEsRepitente}$

AlumnoEsRepitenteNoEncontradoE

$\Xi \text{MisEstudiantes}$

$\text{rep!} : \text{REP}$

$\text{alumno?} \notin \text{dom legajos}$

$\text{rep!} = \text{alumnoNoEncontrado}$

$\text{AlumnoEsRepitenteE} == \text{AlumnoEsRepitenteNoEncontradoE}$

$\text{AlumnoEsRepitenteOk} == \text{AlumnoEsRepitenteSiOk} \vee \text{AlumnoEsRepitenteNoOk}$

$\text{AlumnoEsRepitente} == \text{AlumnoEsRepitenteOk} \vee \text{AlumnoEsRepitenteE}$