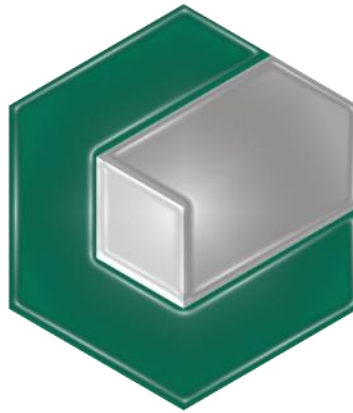


Universidad Tecnológica de La Habana José Antonio Echeverría



Trabajo de Curso de Bases de Datos

“Control de la docencia en un centro estudiantil”

Enunciado 1

Autores:

Lilian Rosa Rojas Rodríguez – lrojas@ceis.cujae.edu.cu

Eduardo Alejandro González Martell - egonzalez@ceis.cujae.edu.cu

Tutor:

Lic. Delvia Noris González Martínez – delvia@tesla.cujae.edu.cu

Grupo docente: IF-21

Carrera: Ing. Informática

La Habana, 2023

Resumen

El objetivo de este proyecto de curso de bases de datos es desarrollar un sistema eficiente para el control de la docencia en un centro estudiantil. La problemática que se aborda es la falta de un sistema integrado que permita a los administradores, secretarios docentes y estudiantes acceder y gestionar información relevante relacionada con la docencia, asignación de grupos y aulas, seguimiento del rendimiento académico, generación del escalafón, control de las asignaturas, gestión de las bajas, los nuevos ingresos y la promoción, entre otros. Para solucionar esta problemática, se diseñó e implementó una base de datos robusta y flexible. Esta base de datos almacena información sobre estudiantes, asignaturas, evaluaciones, entre otros. Se establecieron relaciones entre estas entidades para permitir consultas complejas y proporcionar una visión integral de la docencia. Además, se desarrolló una interfaz de usuario amigable que permite a los usuarios interactuar con la base de datos de manera eficiente. En resumen, este proyecto proporciona una solución integral para el control de la docencia en un centro estudiantil, mejorando la eficiencia y la transparencia en la gestión académica.

Palabras Claves: centro estudiantil, estudiantes, asignaturas, gestión.

Índice de Contenido

Introducción.....	1
1. Requisitos funcionales del proyecto	2
2. Diseño de la base de datos	4
2.1 Listado de entidades identificadas con sus atributos	4
2.2 Listado de relaciones identificadas.....	5
2.3 Representación de entidades y relaciones en el DER	6
2.3.1 Consideraciones generales.....	6
2.3.2 Creación del DER	6
2.4 Modelo lógico global de datos	7
2.5 Modelo físico de la base de datos.....	7
2.6 Diferencias entre el modelo lógico y el modelo físico.....	8
3. Diseño de la aplicación.....	9
3.1 Diagrama de clases y Listado de clases	9
3.2 Seguridad de la aplicación. Roles y privilegios por roles.....	13
3.2.1 Rol de Administrador.....	13
3.2.2 Rol de Secretario Docente	13
3.2.3 Rol de Estudiante	14
3.3 Documentación de la interfaz de la aplicación	14
3.3.1 Consideraciones generales.....	14
3.3.2 Principios y Patrones de diseño de la interfaz.....	15
3.3.3 Documentación de la Interfaz.....	16
3.4 Reportes diseñados:	24
Conclusiones.....	26
Recomendaciones.....	27
Referencias Bibliográficas	28
Anexos	29
Anexo 1: Script de creación de las tablas de la base de datos	29

Índice de Figuras

Figura 1: DER generado por la herramienta ERECASE.	6
Figura 2: DER generado por la herramienta Draw.io.	7
Figura 3: Modelo físico de la base de datos generado con la herramienta Database Designer.....	8
Figura 4: Diagrama de clases UML de las clases DTO de la aplicación generada con la herramienta PlantUML.....	9
Figura 5: Diagrama de clases UML de las clases Services de la aplicación generada con la herramienta PlantUML.....	10
Figura 6: Diagrama de clases UML de clases auxiliares de la aplicación generada con la herramienta PlantUML.....	10
Figura 7: Pantalla de Carga.....	16
Figura 8: Pantalla de Autenticación.....	17
Figura 9: Pantalla de Usuario en Administrador	18
Figura 10: Pantalla de Inicio en Estudiante.	19
Figura 11: Pantalla de Evaluaciones en Estudiante.	19
Figura 12: Pantalla de Escalafón en Estudiante.	20
Figura 13: Pantalla de Alumnos en Secretario Docente.....	21
Figura 14: Pantalla de Bajas en Secretario Docente.....	21
Figura 15: Pantalla de Asignaturas en Secretario Docente.....	22
Figura 16: Pantalla de Evaluaciones en Secretario Docente.....	22
Figura 17: Pantalla de Escalafón en Secretario Docente.	23
Figura 18: Pantalla de Selección de Reportes	23

Introducción

Problemática: En los centros estudiantiles se desea automatizar el control de los resultados docentes alcanzados por los alumnos, así como la actualización de la matrícula y otras informaciones necesarias para controlar la docencia. De los alumnos se conocen su código, nombres y apellidos, sexo y municipio de residencia. Un alumno puede ser promovido al año inmediato superior (si aprobó todas las asignaturas del año que termina); repitente (si tiene una o más asignaturas desaprobadas) o baja, en tal caso se almacena la causa de la baja (enfermedad; traslado; culminación de los estudios; desaprobar por segunda vez un año, para lo cual basta con que desaprobe una asignatura). Un grupo está formado por varios alumnos y un alumno pertenece a solo un grupo. Un grupo está identificado por el año y su número (puede repetirse para años diferentes, pero es único en un mismo año). Para los alumnos de cada grupo se conocen los promedios (anual o acumulado). De cada año se conoce el año y el curso escolar (ejemplo: 2021-2022). En un mismo año se imparten varias asignaturas y una asignatura dada pertenece a un solo año. De las asignaturas se tiene el nombre de la asignatura y la cantidad de horas planificadas. Se conocen las evaluaciones (Excelente, Bien, Aprobado, Desaprobado) obtenidas por cada estudiante de un grupo en cada asignatura. Dado un año y cada grupo de ese año, se obtiene el escalafón del año. Para ello se desarrollará un sistema que estará disponible en cada Facultad del Centro.

Objetivo: Dar solución a la problemática planteada con la creación de un *software* que le permita al personal docente de cada facultad del centro gestionar todo el proceso de gestión y control de la docencia, de forma rápida, directa y sencilla.

Herramientas utilizadas: Para la creación del Diagrama Entidad-Relación, se usó la herramienta ERECASE, la cual permite también la generación automática del script de la base de datos. Además, se empleó el software Draw.io en su versión de extensión para VSCode, para poder generar Diagramas de Entidad-Relación más formales que los que ofrece ERECASE. Por otra parte, se utilizó la aplicación móvil *Database Designer* para modelar el modelo físico de la base de datos. Para poder diseñar el diagrama de clases UML se empleó el *software* PlantUML, en su versión de plugin para IntelliJIDEA. Para poder trabajar con la base de datos en PostgreSQL se utilizaron los *softwares* PGAdmin III y Navicat Premium. Para poder trabajar con la lógica en Java se utilizó Eclipse IDE. Se utilizó el framework Java Swing para diseñar las interfaces visuales. Se empleó la herramienta JasperStudio para el desarrollo de los reportes de la aplicación.

Estructura del documento: El presente documento está estructurado en tres epígrafes principales. En el primer epígrafe se listan los requisitos funcionales. En el segundo y tercer epígrafes se analizan en sus correspondientes sub-epígrafes, diferentes aspectos del diseño de la base de datos y la aplicación. Luego se otorgarán las conclusiones alcanzadas tras la finalización del presente proyecto, las recomendaciones, la bibliografía empleada y los anexos.

1. Requisitos funcionales del proyecto

Los principales requerimientos funcionales del sistema son:

1. Listado de los alumnos por grupo:

En esta salida deben aparecer las siguientes informaciones:

- curso escolar (ejemplo: 2021-2022)
- año
- número del grupo (puede repetirse para años diferentes, pero es único en un mismo año)

y, para cada alumno, por número de orden (que se asigna por orden alfabético de los apellidos):

- número de orden
- nombres y apellidos
- código del alumno

2. Listado de las asignaturas que se imparten en cada año:

En esta salida deben aparecer las siguientes informaciones:

- curso escolar
- año

y para cada asignatura:

- nombre de la asignatura
- cantidad de horas

3. Listado de las evaluaciones por grupo en cada asignatura:

En esta salida deben aparecer las siguientes informaciones:

- curso escolar
- año
- número del grupo
- nombre de la asignatura

y, para cada alumno, por número de orden:

- número de orden
- nombres y apellidos
- evaluación obtenida en esa asignatura (Excelente, Bien, Aprobado, Desaprobado)

4. Listado de los promedios finales obtenidos por los alumnos en cada grupo:

En esta salida se desea obtener la siguiente información (anual o acumulado):

- período que se reporta (anual o acumulado)
- año (si es anual) o periodo (intervalo de tiempo)
- número del grupo

y, para cada alumno, por número de orden:

- número de orden
- nombres y apellidos
- promedio (anual o acumulado)

5. Escalafón:

En esta salida se desea obtener por grupo y para todo el año. Debe contener las informaciones siguientes:

- curso escolar

- año
 - número del grupo (si se desea para un solo grupo; si es para todo el año, no aparece)
- y, para cada alumno, según el orden de escalafón:
- número de escalafón del año
 - número de escalafón del grupo
 - nombres y apellidos
 - promedio acumulado
 - número de orden
 - sexo
 - municipio de residencia
6. **Certificación de notas de cada estudiante:**
 Las informaciones que debe contener esta salida son:
- nombres y apellidos
- y, para cada año académico:
- curso escolar
 - nombre de la asignatura
 - calificación obtenida (Excelente, Bien, Aprobado, Desaprobado)
 - promedio docente
7. **Listado de los alumnos desaprobados por grupo para un rango de fecha:**
 Este listado debe contener:
- curso escolar
 - año
 - número del grupo
- y, para cada estudiante desaprobado del grupo:
- número de orden
 - listado de asignaturas desaprobadas
8. **Listado, por grupo o por año, de los alumnos que causan baja en un curso:**
 Este listado debe contener:
- curso escolar
 - año
 - número del grupo (si es para un grupo)
- y, para cada alumno que causa baja, por número de orden:
- número de orden
 - nombres y apellidos
 - grupo (si es para un año)
 - causa de la baja
9. **Listado de los repitentes por año:**
 Debe contener la siguiente información:
- curso escolar
 - año
- y, para cada alumno repitente, por número de orden:
- número de orden
 - nombres y apellidos
 - grupo

10. **CRUD** de las principales entidades.
11. Realizar eficientemente la promoción de los estudiantes al final del curso.

2. Diseño de la base de datos

2.1 Listado de entidades identificadas con sus atributos

Las entidades identificadas con sus atributos son:

1. **Student:** Esta entidad constituye una generalización que se refiere a los Estudiantes, cuenta con los siguientes atributos:
 - **id_std:** identificador único del estudiante.
 - **names_std:** nombres del estudiante.
 - **lastnames_std:** apellidos del estudiante.
 - **sex_std:** sexo del estudiante.
 - **mun_std:** municipio del estudiante.

Las especializaciones de la generalización Student son:

- **Promoted:** En referencia a los estudiantes promovidos.
 - **Repeater:** En referencia a los estudiantes repitentes.
 - **Dismissal:** En referencia a los estudiantes baja, de los cuales se sabe la causa de la baja.
2. **Academic_Subject:** Esta entidad se refiere a las asignaturas del curso y cuenta con los siguientes atributos:
 - **id_sub:** identificador único de la asignatura.
 - **name_sub:** nombre de la asignatura.
 - **hours_plan_sub:** cantidad de horas dedicadas a la asignatura.
 3. **Evaluation:** Constituye una agregación que contiene a la relación existente entre las entidades Student y Academic_Subject, y una evaluación que se expresa en la relación con la entidad Type_Of_Evaluation.
 4. **Type_Of_Evaluation:** Nomenclador que contiene las evaluaciones posibles que se pueden obtener y cuenta con los siguientes atributos:
 - **id_eval:** identificador único de la evaluación.
 - **eval:** nombre de la evaluación.
 - **eval_num:** número de la evaluación.
 5. **Cause_Of_Dismissal:** Nomenclador que contiene las posibles causas de baja que se tienen y cuenta con los siguientes atributos:
 - **id_dm:** identificador único de la causa de baja.
 - **dm:** causa de baja.
 6. **Academic_Year:** Esta entidad se refiere al año que se cursa y cuenta con los siguientes atributos:
 - **id_y:** identificador único del año.
 - **year_y:** año que se cursa, ya sea primero, segundo, tercero o cuarto año.
 - **school_course_start_y:** año de inicio del curso escolar.
 - **school_course_end_y:** año de inicio del curso escolar.
 7. **Group_St:** Esta entidad se refiere al grupo de los estudiantes, es una entidad débil que se identifica completamente teniendo en cuenta el año y su id, y cuenta con los siguientes atributos:
 - **id_gr:** identificador del grupo.

- **gr_number**: número del grupo.
- **classroom_gr**: número del aula en la que radica el grupo.

2.2 Listado de relaciones identificadas

Las relaciones identificadas son:

- **CD_DI**: Entre la entidad **CAUSE_OF_DISMISSAL** y la entidad **DISMISSAL** existe una relación de 1 a muchos debido a que una causa de baja se puede corresponder con varias bajas, pero una baja solo puede corresponderse con una de las siguientes causas de baja: enfermedad, traslado, culminación de estudios y desaprobado por segunda vez un año. De esta manera el extremo 1 se corresponde con la entidad **CAUSE_OF_DISMISSAL** y el extremo muchos con la entidad **DISMISSAL**.
- **ST_GR**: Existe una relación de 1 a muchos entre las entidades **GROUP_ST** y **STUDENT** porque un grupo está formado por varios alumnos y un alumno solo puede pertenecer a un grupo. En esta relación el extremo 1 se corresponde con la entidad **GROUP_ST** mientras que el extremo muchos se corresponden con la entidad **STUDENT**. Esta relación además presenta los atributos **anuales_avg** (promedio anual) y **ac_avg** (promedio acumulado).
- **SU_Y**: Existe una relación de 1 a muchos entre las entidades **ACADEMIC_YEAR** y **ACADEMIC_SUBJECT** debido a que en un año se imparten varias asignaturas y una asignatura dada solo pertenece a un año. En esta relación el extremo 1 se corresponde con la entidad **ACADEMIC_YEAR** mientras que el extremo mucho se corresponde con la entidad **ACADEMIC_SUBJECT**.
- **E_TE**: Entre la entidad **EVALUATION** y la entidad **TYPE_OF_EVALUATION** se evidencia una relación de 1 a muchos debido a que un tipo de evaluación se puede corresponder con varias evaluaciones, pero una evaluación solo puede corresponderse con uno de los siguientes tipos de evaluación: Excelente, Bien, Aprobado y Desaprobado. De esta manera el extremo 1 se corresponde con la entidad **TYPE_OF_EVALUATION** y el extremo muchos con la entidad **EVALUATION**.
- **Y_GR**: Entre la entidad **ACADEMIC_YEAR** y la entidad **GROUP_ST** existe una relación de 1 a muchos porque en un mismo año existen diferentes grupos y un grupo dado solo puede pertenecer a un año. En esta relación el extremo 1 se corresponde con la entidad **ACADEMIC_YEAR** mientras que el extremo muchos se corresponden con la entidad **GROUP_ST**. Esta relación permite identificar la entidad débil **GROUP_ST** con **ACADEMIC_YEAR**.
- **ST_SU**: Existe una relación de muchos a muchos entre las entidades **STUDENT** y **ACADEMIC_SUBJECT** debido a que un alumno puede cursar más de una asignatura y una asignatura puede ser cursada por muchos alumnos.

2.3 Representación de entidades y relaciones en el DER

2.3.1 Consideraciones generales

Las consideraciones generales son:

- Las entidades **TYPE_OF_EVALUATION** y **CAUSE_OF_DISMISSAL** son nomencladores.
- El atributo llave **id_gr** de la entidad **GROUP_ST** coincide con el número del grupo con el cual se identifica dentro de un mismo año.
- Los promedios anuales y acumulados y el escalafón son atributos calculables a partir de las evaluaciones de las asignaturas, por lo que se decidió no incluirlas en el diagrama.

2.3.2 Creación del DER

El Diagrama Entidad-Relación generado por la herramienta ERECASE fue el siguiente:

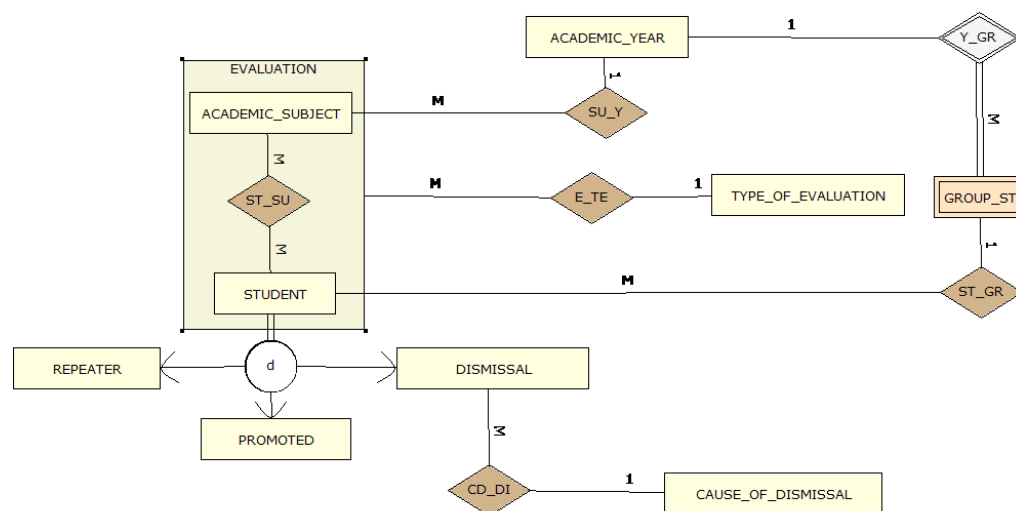


Figura 1: DER generado por la herramienta ERECASE.

El Diagrama Entidad-Relación generado con la herramienta Draw.io, es el siguiente:

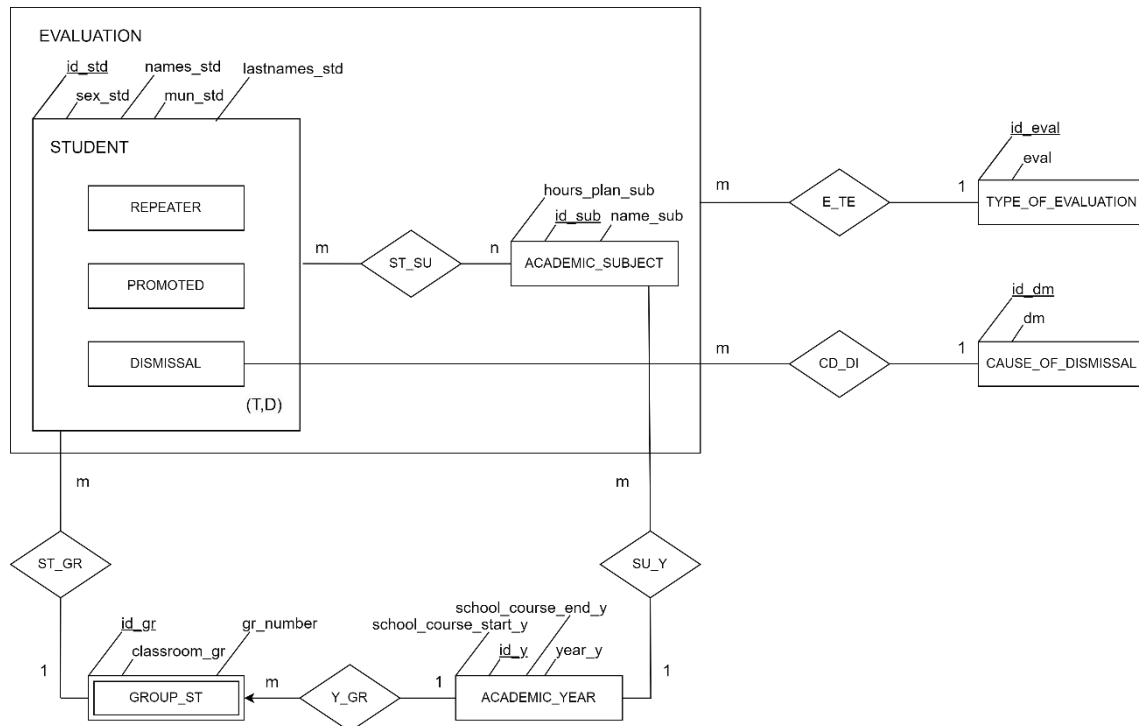


Figura 2: DER generado por la herramienta Draw.io.

Nota: Las llaves como **id_gr**, no se ven correctamente al subrayarlas.

2.4 Modelo lógico global de datos

Luego del proceso de normalización se presenta a continuación el modelo lógico global de datos resultante:

EVALUATION(id_sub, id_std, id_eval)

TYPE_OF_EVALUATION(id_eval, eval)

ACADEMIC_SUBJECT(id_sub, name_sub, hours_plan_sub, id_y)

ACADEMIC_YEAR(id_y, year_y, school_course_start_y, school_course_end_y)

GROUP_ST(id_gr, id_y, gr_number, classroom_gr)

STUDENT(id_std, names_std, lastnames_std, sex_std, mun_std, id_gr, id_y)

REPEATER(id_std)

PROMOTED(id_std)

DISMISSAL(id_std, id_dm)

CAUSE_OF_DISMISSAL(id_dm, dm)

2.5 Modelo físico de la base de datos

A continuación, se presenta el modelo físico de la base de datos:

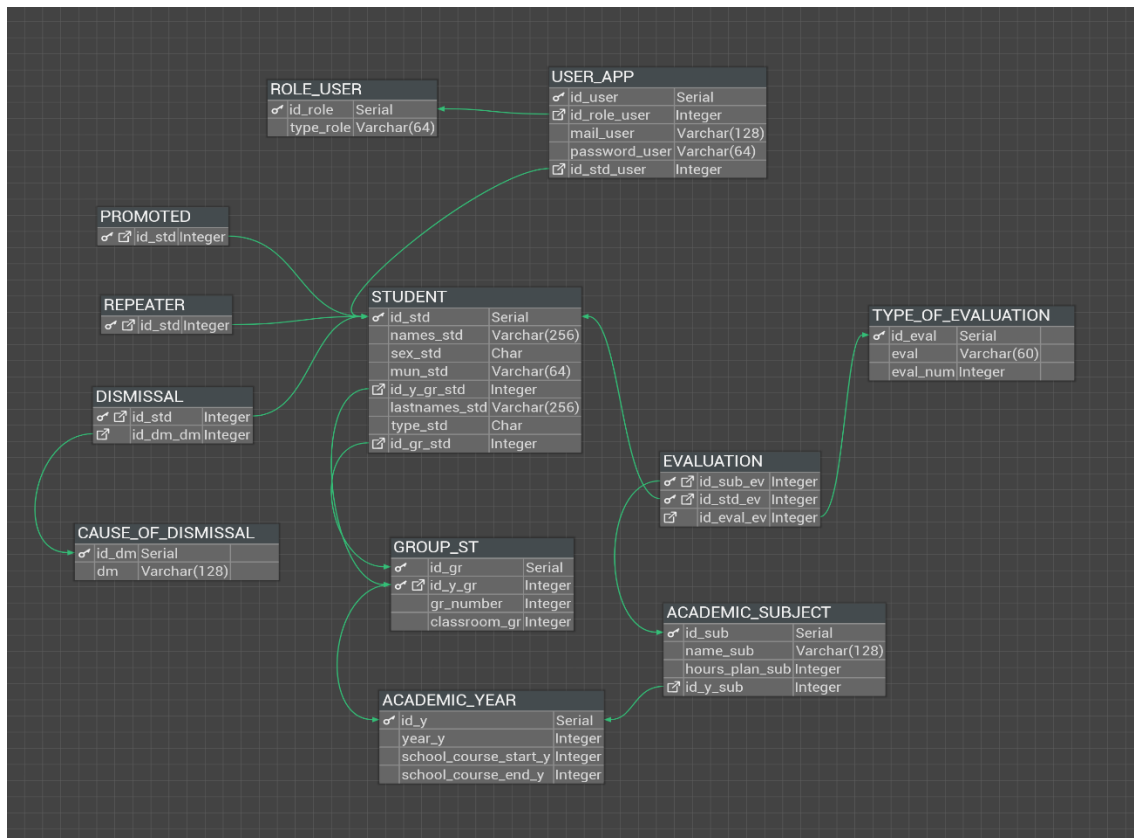


Figura 3: Modelo físico de la base de datos generado con la herramienta Database Designer.

2.6 Diferencias entre el modelo lógico y el modelo físico

Las diferencias más características detectadas entre el modelo lógico global de datos y el modelo físico de la base de datos son las siguientes:

- En el modelo lógico se trata con entidades y atributos, en el físico se trata con tablas y columnas.
- Los nombres de los atributos en el modelo físico varían ligeramente con respecto al modelo lógico, sobre todo en los casos de las llaves foráneas, lo cual permite diferenciarlas luego a través de las diferentes tablas donde se relacionan.
- En el modelo físico se presentan los tipos de datos de cada atributo.
- El modelo físico contiene las entidades usuario y rol de usuario para poder otorgar o denegar ciertos privilegios dentro de la aplicación. (**ROLE_USER** es un nomenclador que permite clasificar los distintos roles de la aplicación, y **USER_APP** contiene la información de los usuarios de la aplicación).
- En el modelo físico la entidad estudiante presenta el tipo de estudiante para facilitar ciertas operaciones.

3. Diseño de la aplicación

3.1 Diagrama de clases y Listado de clases

A continuación, se muestran los diagramas de clases UML de las clases presentes en la aplicación:

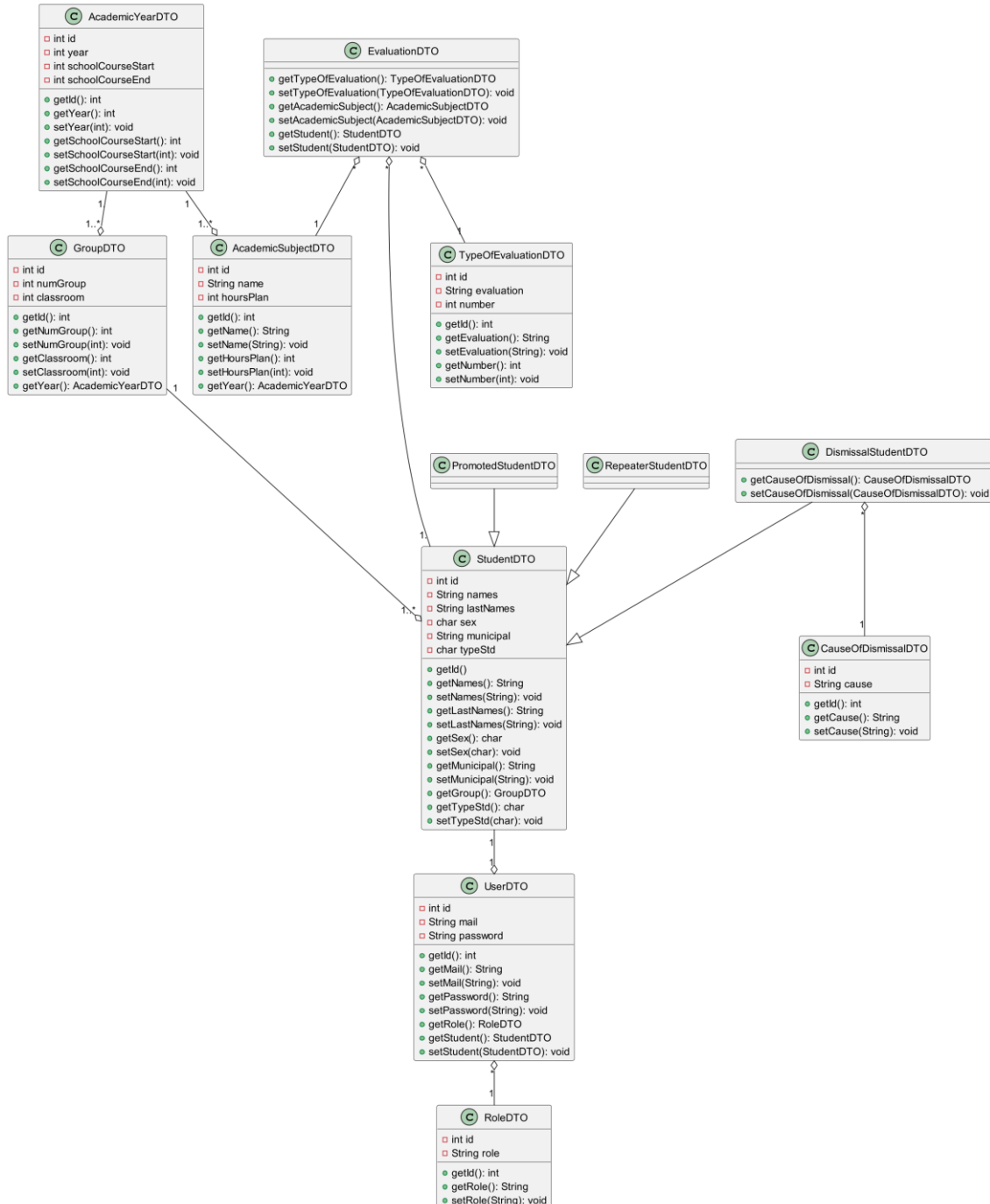


Figura 4: Diagrama de clases UML de las clases DTO de la aplicación generada con la herramienta PlantUML.

conteniendo los atributos y los gets y sets correspondientes. Las clases DTO diseñadas son:

- **AcademicSubjectDTO:** Clase que modela una asignatura, con los mismos atributos de la tabla **ACADEMIC_SUBJECT**, pero con la particularidad de que contiene como atributo además el año académico al que pertenece la asignatura, almacenado como **AcademicYearDTO**.
- **AcademicYearDTO:** Clase que modela un año académico, con los mismos atributos de la tabla **ACADEMIC_YEAR**.
- **CauseOfDismissalDTO:** Clase que modela el nomenclador causa de baja, con los mismos atributos que la tabla **CAUSE_OF_DISMISSAL**.
- **DismissalStudentDTO:** Clase que modela un estudiante baja, con los mismos atributos de la tabla **DISMISSAL**, pero con la particularidad de que contiene como atributo además la causa de baja que tiene el estudiante, almacenado como **CauseOfDismissalDTO**. Esta clase hereda de la clase **StudentDTO**.
- **EvaluationDTO:** Clase que modela las evaluaciones de un estudiante en una asignatura con una nota específica, con los mismos atributos de la tabla **EVALUATION**, pero con la particularidad de que contiene como atributo además al estudiante, la asignatura, y la evaluación, almacenados como **StudentDTO**, **AcademicSubjectDTO** y **TypeOfEvaluationDTO** respectivamente.
- **GroupDTO:** Clase que modela un grupo, con los mismos atributos de la tabla **GROUP_ST**, pero con la particularidad de que contiene como atributo además el año académico al que pertenece el grupo, almacenado como **AcademicYearDTO**.
- **PromotedStudentDTO:** Clase que modela un estudiante promovido, con los mismos atributos que la tabla **PROMOTED**. Esta clase hereda de la clase **StudentDTO**.
- **RepeaterStudentDTO:** Clase que modela un estudiante repitente, con los mismos atributos que la tabla **REPEATER**. Esta clase hereda de la clase **StudentDTO**.
- **RoleDTO:** Clase que modela el nomenclador rol de usuario, con los mismos atributos que la tabla **ROLE_USER**.
- **StudentDTO:** Clase que modela un estudiante, con los mismos atributos que la tabla **STUDENT**, pero con la particularidad de que contiene como atributo además el grupo al que pertenece el estudiante, almacenado como **GroupDTO**. Esta clase es abstracta, y de ella heredan todos los tipos de estudiantes.

- **TypeOfEvaluationDTO:** Clase que modela el nomenclador tipo de evaluación, con los mismos atributos que la tabla **TYPE_OF_EVALUATION**.
- **UserDTO:** Clase que modela un usuario, con los mismos atributos que la tabla **USER_APP**, pero con la particularidad de que contiene como atributo además el rol que tiene el usuario, almacenado como **RoleDTO**, y el estudiante, en caso de ser de ese rol, almacenado como **StudentDTO**.
- **Clases Services:** Clases que contienen toda la lógica de negocio de la aplicación. Las clases services diseñadas son:
 - **AcademicSubjectServices:** Clase que modela los servicios referentes al trabajo con la tabla **ACADEMIC_SUBJECT**.
 - **AcademicYearServices:** Clase que modela los servicios referentes al trabajo con la tabla **ACADEMIC_YEAR**.
 - **AuthServices:** Clase que modela los servicios referentes al sistema de autenticación del login de la aplicación.
 - **CauseOfDismissalServices:** Clase que modela los servicios referentes al trabajo con la tabla **CAUSE_OF_DISMISSAL**.
 - **EvaluationServices:** Clase que modela los servicios referentes al trabajo con la tabla **EVALUATION**.
 - **GroupServices:** Clase que modela los servicios referentes al trabajo con la tabla **GROUP_ST**.
 - **ReportServices:** Clase que modela los servicios referentes a los diferentes reportes de la aplicación.
 - **RoleServices:** Clase que modela los servicios referentes al trabajo con la tabla **ROLE_USER**.
 - **StudentServices:** Clase que modela los servicios referentes al trabajo con la tabla **STUDENT**.
 - **TypeOfEvaluationServices:** Clase que modela los servicios referentes al trabajo con la tabla **TYPE_OF_EVALUATION**.
 - **UserServices:** Clase que modela los servicios referentes al trabajo con la tabla **USER_APP**.
- **Clases Auxiliares:** Algunas de las clases auxiliares que complementan el trabajo con las clases principales. Algunas de las clases auxiliares diseñadas son:
 - **FunctionBuilder:** Contiene los métodos para poder construir funciones de consulta, lo que facilita el trabajo con la base de datos.
 - **FunctionType:** Nomenclador que contiene los principales tipos de funciones CRUD empleadas en el constructor de funciones **FunctionBuilder**.

- **Encription:** Permite la codificación y decodificación de mensajes a través del estándar MD5. Se emplea principalmente en la codificación y decodificación de contraseñas.

Las consideraciones hechas al abordar la solución se resumen en:

- Las clases que siguen el patrón DTO se diseñaron con el mismo conjunto de atributos y la misma disposición que en la base de datos; aunque se realizaron algunas modificaciones como el uso de agregaciones y herencias que facilitan el trabajo con los datos usando los principios de la Programación Orientada a Objetos (POO).
- No se generaron métodos sets para los identificadores de las clases DTO, puesto que los mismos son inmutables.
- Las clases de servicios referentes al trabajo con las tablas de la base de datos, solamente contienen un número limitado de funcionalidades CRUD, para así aumentar la seguridad e integridad de la base de datos.

3.2 Seguridad de la aplicación. Roles y privilegios por roles

Como parte de la seguridad de la aplicación se definieron 3 roles principales: *Administrador*, *Secretario Docente* y *Estudiante*, todos con privilegios bien definidos.

3.2.1 Rol de Administrador

El administrador solo tiene acceso a los siguientes privilegios:

- Gestión de usuarios (crear y eliminar usuarios con sus roles).

3.2.2 Rol de Secretario Docente

El secretario docente solo tiene acceso a los siguientes privilegios:

- Ver la información general de todos los alumnos, pudiendo agregar, modificar y eliminar alumnos. Esto también incluye dar de baja a los alumnos.
- Ver la información general de las asignaturas
- Ver la información general de las evaluaciones de todos los estudiantes en todas las asignaturas, pudiendo modificarlas.
- Ver el escalafón tanto a nivel de grupo como de año.
- Ver el listado de los promedios finales obtenidos por los alumnos de cada grupo.
- Ver la certificación de notas de cada estudiante.
- Ver el listado de los alumnos desaprobados por grupo para un rango de fechas (cursos escolares).
- Ver el listado de alumnos que causan baja en un curso, por grupo o por año.

- Ver el listado de los repitentes por año.

3.2.3 Rol de Estudiante

El estudiante solo tiene acceso a los siguientes privilegios:

- Ver su información general, como sus nombres y apellidos, su municipio, su sexo, su grupo, su año, su curso escolar, su estado y su promedio.
- Ver sus evaluaciones en las diferentes asignaturas de los diferentes años.
- Ver el escalafón tanto a nivel de grupo como de año.

3.3 Documentación de la interfaz de la aplicación

3.3.1 Consideraciones generales

De forma general, se tuvieron las siguientes consideraciones a la hora de diseñar la interfaz gráfica:

- **Fuentes:** Para una mayor homogeneidad se empleó como fuente, en todo el programa, Roboto, en todas sus variantes (negrita, cursiva y plano). Se empleó la misma fuente que es muy empleada en las aplicaciones hoy en día por su simplicidad, facilidad de lectura y vistosidad. El tamaño de la fuente fue variable, en dependencia de la situación: para los encabezados se empleó un tamaño de entre 20 y 28 puntos; para la representación de la información en tablas se empleó un tamaño de 15 puntos; y para la representación de la información en general, se utilizó un tamaño de entre 15 y 18 puntos. El color de la fuente es negro, exceptuando en casos que, por ejemplo, es roja porque hay un error en un campo específico, o se decidió otro color para mejorar el contraste con el fondo.
- **Tablas:** Para el mostrado de la información principal se emplearon las tablas de información. Las mismas fueron programadas para que se pudieran reordenar en dependencia de la columna seleccionada por el usuario, tanto de manera ascendente como descendente. Algunas tablas funcionan como un directorio de alumnos, como es el caso de la de los listados de alumnos, en las cuales, con solo un doble click en el paciente deseado, se podrá acceder a su información e incluso editarla según sea el caso. Para la solución informática de este problema se decidió que las celdas no fueran modificables, puesto que esto podría conllevar a errores en el futuro.
- **Iconografía:** Se utilizó como biblioteca de iconos, la biblioteca gráfica en línea icons8.com. Se siguió un diseño de iconos simplista e intuitivo para no complejizar la visual del programa.
- **Imágenes:** Se emplearon imágenes en ciertas secciones del programa por diferentes motivos. Por ejemplo, en la sección de la pantalla de carga para mostrar el logo de la aplicación; y en la sección de autenticación, para el diseño de la interfaz, y como avatar personalizado.
- **Reportes de Salida:** Los reportes de la aplicación se muestran a través de las tablas. Las mismas contienen la información necesaria para la comprensión del significado del reporte.

- **Componentes utilizados. Prevención y Tratamiento de errores:** Para minimizar al máximo la introducción de errores se emplearon los componentes indicados para cada tipo de dato: listas de desplegables, cuadros de edición validados y formateados para cada escenario, y selecciones múltiples. En caso de que el usuario no introduzca todos los datos necesarios, o no estén correctamente llenados, se lanza un mensaje de error, y se indican los campos que se requieren modificar para el correcto funcionamiento del sistema de guardado.

De forma general, la aplicación se encuentra documentada con ayuda de cómo usar el sistema en las diferentes secciones. Las pantallas están desarrolladas en forma rectangular horizontal y con las siguientes dimensiones (diseño desktop):

- **Pantalla de carga:** 512x385 (4:3).
- **Autenticación:** 800x512 (39:25).
- **App Principal:** 1200x675 (16:9).
- **Diálogos generales:** 700x500 (7:5)

3.3.2 Principios y Patrones de diseño de la interfaz

La interfaz del proyecto se enfoca en el cumplimiento de buenas prácticas y estándares de diseño de interfaces gráficas de usuario en aplicaciones informáticas. Propicia una fácil comprensión del usuario. Las operaciones de interfaz son fáciles de localizar e iniciar, se economizan las acciones que tiene que realizar el usuario. La estética ayuda a la comprensión de cada funcionalidad. Se han organizado las operaciones jerárquicas de una manera que minimice la profundidad a la que un usuario debe navegar y la aplicación está libre de errores ortográficos.

Se ponen de manifiesto las reglas de oro puesto que el usuario tiene en todo momento control sobre la aplicación, se contribuye a reducir su carga de memoria y la interfaz es consistente. El usuario recibe respuesta por cada acción que realiza. Los mecanismos de interacción, íconos y procedimientos son consistentes. El sistema se anticipa a la comisión de errores y ayuda al usuario a corregirlos al introducir datos y se manejan las excepciones para que no deje de funcionar el programa cuando se cometen errores.

Cada componente del programa tiene un nombre con carácter descriptivo que permite comprender la información contenida en él y que es único. Los datos similares dentro de una misma ventana están agrupados visualmente, alineados y existen etiqueta para nombrar grupos de datos similares.

Se decidió seguir el patrón de diseño *Dashboard* en la pantalla principal del sistema pues es un patrón muy empleado en la actualidad. Se utiliza un diseño basado en el *Material Design*. Se utilizan fondos claros para las áreas principales de las pantallas que contrastan con la letra. También se utilizan imágenes, íconos y una especie de gráficos (tarjetas) en la pantalla principal para mejor experiencia de usuario y mejorar.

Se usan las cajas de diálogo modales para obligar al usuario a realizar una determinada tarea antes de que pueda continuar el trabajo en cualquier otro

cuadro o ventana. Se alerta a los usuarios de las consecuencias de sus acciones que pueden producir errores o que pueden llevar a estados indeseados.

Los principales patrones de diseño empleados en la aplicación son:

- **Patrón de organización de página. Secciones tituladas:** Se emplea para agrupar grandes cantidades de información, ordenándolos en secciones de igual temática, con un título sugerente. Se pone de manifiesto en cada sección del programa donde se agrupan las informaciones.
- **Patrón de navegación. Panel Modal:** Se emplea para obligar el cumplimiento de una tarea por parte del usuario, marcando pautas en el flujo de trabajo. Se pone de manifiesto en todas las ventanas emergentes del programa.
- **Patrón de Hacer. Botón “Aceptar”:** Se emplea para demarcar visualmente un botón para aceptar y finalizar correctamente el flujo de trabajo actual. Se pone de manifiesto en todas las secciones del programa que necesiten del mismo.
- **Patrón para obtener información del usuario. Valores predeterminados:** Se emplean para reducir la carga de trabajo del usuario, así como prevenir posibles errores. Se pone de manifiesto en cada sección del programa donde se otorguen la selección de valores predeterminados.
- **Patrón de organización de contenido. Buscar:** Se emplea para el filtrado de vastas listas de contenidos, ordenando el resultado en dependencia de la preferencia del usuario.

3.3.3 Documentación de la Interfaz

Pantalla de Carga

Permite calmar la impaciencia del usuario por la demora del inicio del sistema, y a poder a la misma vez hacer tareas de inicialización de la interfaz como conectarse a la base de datos.

Como aspectos a destacar, contiene un panel de animación desarrollado internamente, y una barra de progreso que en caso de que llegue al final y detecte que no se ha cargado el sistema, la misma vuelve al inicio.



Figura 7: *Pantalla de Carga*

Pantalla de Autenticación

Permitirá restringir el ingreso a la aplicación, además de poder asignar privilegios a los usuarios en dependencia de su rol.

Contiene campos de edición validados en tamaño, un botón para ocultar y mostrar la contraseña y un botón para ingresar. Además, al igual que sucede en la pantalla principal, se desarrolló una barra superior personalizada semitransparente, y con las funcionalidades habituales de este tipo de componentes: la movilidad de la ventana, minimizar la ventana y cerrar la aplicación. Si ocurre algún error en la autenticación se muestra el mismo a través de un panel flotante con animación (Toast Panel), y si se logra autenticar, se le da la bienvenida al sistema al usuario a través de un cuadro de diálogo.



Figura 8: Pantalla de Autenticación

Pantallas Principales de Administrador

Será la pantalla principal en que los usuarios con el rol de Administrador interactuarán en todo momento con la aplicación. Mostrará a modo de secciones las diferentes funcionalidades del programa que le corresponden al Administrador. Las interfaces gráficas principales de este apartado son:

- **Pantalla de Usuario:** En esta pantalla el Administrador podrá gestionar los usuarios, pudiendo realizar diferentes operaciones como ver los usuarios (pudiendo filtrarlos por diferentes campos) y dar de alta o de baja en el sistema a usuarios. Ver **Figura 9**.

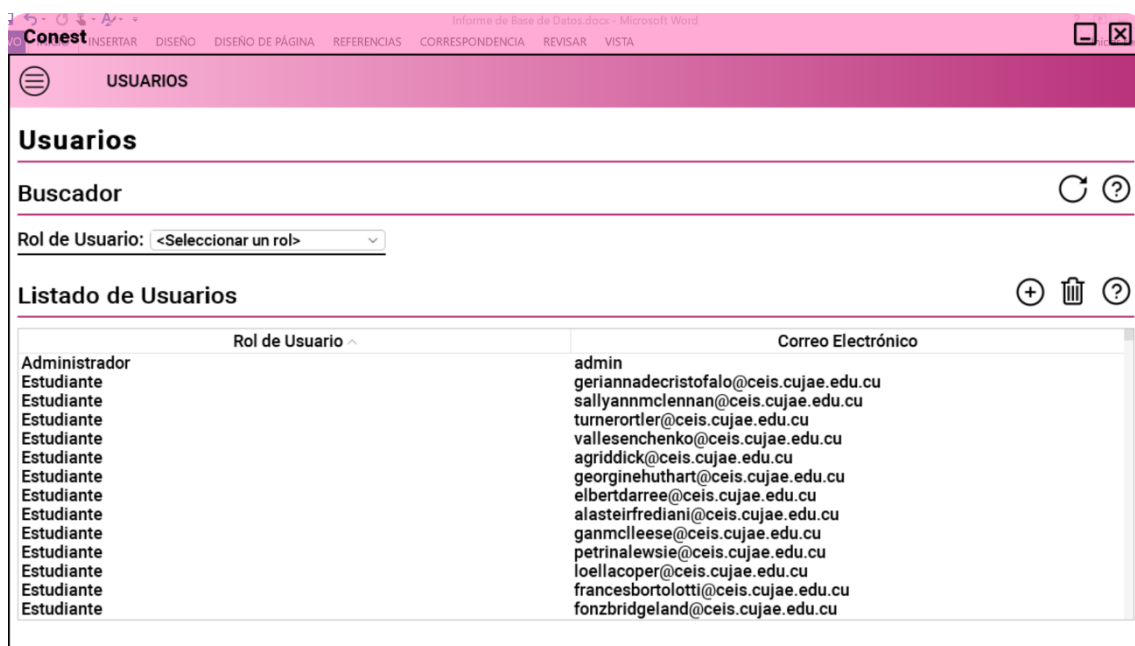


Figura 9: Pantalla de Usuario en Administrador

Pantallas Principales de Estudiante

Será la pantalla principal en que los usuarios con el rol de Estudiante interactuarán en todo momento con la aplicación. Mostrará a modo de secciones las diferentes funcionalidades del programa que le corresponden al Estudiante. Las interfaces gráficas principales en este apartado son:

- **Pantalla de Inicio:** En esta pantalla el Estudiante puede ver su información general con datos como sus nombres y apellidos, su municipio, su estado y su promedio general acumulado. Ver **Figura 10**.
- **Pantalla de Evaluaciones:** En esta pantalla el Estudiante puede ver sus evaluaciones en las diferentes asignaturas en los diferentes años, pudiendo filtrarlas por distintos campos. Ver **Figura 11**.
- **Pantalla de Escalafón:** En esta pantalla el Estudiante puede ver su posición en el escalafón, tanto a nivel de grupo, como a nivel de año. Ver **Figura 12**.

Conest INFORME DE BASE DE DATOS.docx - Microsoft Word

INICIO

Información General

Nombre del Estudiante: Gerianna De Cristofalo

Municipio: Boyeros Sexo: Femenino

Curso Escolar: 2023-2023 Año: Primero Grupo: 11

Estado: Promovido

3.93

Figura 10: Pantalla de Inicio en Estudiante.

Conest INFORME DE BASE DE DATOS.docx - Microsoft Word

EVALUACIONES

Evaluaciones

Buscador

Año: <Seleccionar un año> Calificación: <Seleccionar una calificación>

Listado de Evaluaciones

Año	Curso Escolar	Asignatura	Calificación
Primero	2023-2023	Introducción a la Informática	Excelente
Primero	2023-2023	Introducción a la Programación	Excelente
Primero	2023-2023	Matemática Discreta	Aprobado
Primero	2023-2023	Filosofía y Sociedad	Aprobado
Primero	2023-2023	Matemática I	Aprobado
Primero	2023-2023	Inglés 1	Bien
Primero	2023-2023	Educación Física 1	Aprobado
Primero	2023-2023	Historia de Cuba	Bien
Primero	2023-2023	Diseño y Programación Orientada...	Bien
Primero	2023-2023	Matemática Computacional	Aprobado
Primero	2023-2023	Economía Política	Aprobado
Primero	2023-2023	Matemática II	Excelente
Primero	2023-2023	Diseño de Interfaces y Pruebas	Excelente
Primero	2023-2023	Inglés 2	Bien

Figura 11: Pantalla de Evaluaciones en Estudiante.

Informe de Base de Datos.docx - Microsoft Word

Conest INSERTAR DISEÑO DISEÑO DE PÁGINA REFERENCIAS CORRESPONDENCIA REVISAR VISTA

ESCALAFÓN

Escalafón

Seleccionar alcance del Escalafón: ☒ Grupo ☐ Año

Listado de Estudiantes

No. ^	No. Grupo	Nombre y Apellidos	Sexo	Municipio	Promedio Acumulado
1	8	Elbert Darree	Masculino	Plaza de la Revolución	4.33
2	25	Karrie Nern	Femenino	Arroyo Naranjo	4.33
3	2	Simone Bockh	Femenino	Boyeros	4.26
4	4	Cornie Bradfield	Masculino	Boyeros	4.26
5	7	Loella Coper	Femenino	Arroyo Naranjo	4.26
6	26	Saidee Nussey	Femenino	Arroyo Naranjo	4.26
7	14	Burton Hendrickson	Masculino	10 de Octubre	4.2
8	16	Jule Issacov	Masculino	Bauta	4.2
9	23	Gan McLeese	Masculino	Centro Habana	4.2
10	24	Karilynn Metheringham	Femenino	Centro Habana	4.2
11	28	Gonzalo Poynter	Masculino	Plaza de la Revolución	4.2
12	30	Artur Scarman	Masculino	Plaza de la Revolución	4.2
13	32	Kayley Trowbridge	Femenino	La Lisa	4.2
14	27	Turner Ortler	Masculino	Mayarí	4.13
15	29	Ag Riddick	Femenino	Bauta	4.13
16	17	Weber Jeannot	Masculino	Habana del Este	4.06

Figura 12: Pantalla de Escalafón en Estudiante.

Pantallas Principales de Secretario Docente

Será la pantalla principal en que los usuarios con el rol de Secretario Docente interactuarán en todo momento con la aplicación. Mostrará a modo de secciones las diferentes funcionalidades del programa que le corresponden al Secretario Docente. Las interfaces gráficas principales en este apartado son:

- **Pantalla de Alumnos:** En esta pantalla el Secretario Docente puede ver el listado de los diferentes alumnos por grupo. Además, puede realizar distintas operaciones sobre los alumnos como es agregar, modificar y dar baja a los alumnos. Ver **Figura 13**.
- **Pantalla de Bajas:** En esta pantalla el Secretario Docente puede ver el listado de los diferentes alumnos que causan baja por grupo y por año. Además se pueden eliminar del sistema las informaciones de aquellos alumnos bajas que el secretario desee. Ver **Figura 14**
- **Pantalla de Asignaturas:** En esta pantalla el Secretario Docente puede ver el listado de las asignaturas que se imparten en cada año. Ver **Figura 15**.
- **Pantalla de Evaluaciones:** En esta pantalla el Secretario Docente puede ver el listado de las evaluaciones por grupo de cada asignatura de cada alumno, pudiendo agregar dichas evaluaciones. Ver **Figura 16**.
- **Pantalla de Escalafón:** En esta pantalla el Secretario Docente puede ver el listado de estudiantes ordenados en el Escalafón, tanto a nivel de grupo como a nivel de año. Ver **Figura 17**.

- **Pantalla de Selección de Reportes:** En esta pantalla el Secretario Docente puede seleccionar entre los diferentes reportes a los que tiene privilegio. Ver Figura 18.

Alumnos

Seleccionar Grupo

Año: <Seleccionar Año> Grupo: <Seleccionar Grupo>

Listado de Alumnos

No.	Nombre y apellidos	Sexo	Municipio
1	EA	M	Boyeros
2	LB	F	10 de octubre
3	LC	F	Boyeros
4	ED	M	10 de octubre
5	EF	M	Boyeros
6	EG	M	10 de octubre
7	EH	M	Boyeros
8	LI	F	10 de octubre
9	LK	F	Boyeros
10	LM	F	10 de octubre
11	EN	M	Boyeros
12	EO	M	10 de octubre
13	LP	F	Boyeros
14	LQ	F	10 de octubre

Figura 13: Pantalla de Alumnos en Secretario Docente.

BAJAS

Buscador

Causa de baja: <Seleccionar una causa de baja>

Listado de Alumnos

Año	Último Grupo	Nombre y Apellidos	Municipio	Causa de Baja
Primero	11	Lowell Candie	Habana del Este	Enfermedad
Primero	11	Kaycee Skews	Playa	Enfermedad
Primero	12	Cyndi Shannahan	La Lisa	Traslado
Primero	12	Maryrose Beasley	Mayarí	Traslado
Segundo	21	Curr Bartczak	Plaza de la Revolución	Culminación de estudios
Segundo	21	Niels Dudenie	Arroyo Naranjo	Culminación de estudios
Segundo	22	Lelia Sagrott	10 de Octubre	Enfermedad
Segundo	22	Tanhya Jerome	Isla de la Juventud	Enfermedad
Segundo	23	Sampson Camocke	Mayarí	Enfermedad
Segundo	21	Amaleta Martinets	10 de Octubre	Culminación de estudios
Tercero	31	Raffaello Puttnam	La Lisa	Enfermedad
Tercero	31	Linoel Edwick	Plaza de la Revolución	Traslado
Tercero	32	Catlee Headingham	Bauta	Traslado
Tercero	32	Cathi Tuplin	Playa	Traslado

Figura 14: Pantalla de Bajas en Secretario Docente.

Informe de Base de Datos.docx - Microsoft Word

Conest INSERTAR DISEÑO DISEÑO DE PÁGINA REFERENCIAS CORRESPONDENCIA REVISAR VISTA

ASIGNATURAS

Asignaturas

Buscador

Curso Escolar: <Seleccionar un curso escolar> Año: <Seleccionar un año>

Listado de Asignaturas

Curso Escolar ^	Año	Nombre	Cantidad de horas
2020-2021	Primero	Introducción a la Informática	48
2020-2021	Primero	Introducción a la Programación	64
2020-2021	Primero	Matemática Discreta	48
2020-2021	Primero	Filosofía y Sociedad	64
2020-2021	Primero	Matemática I	96
2020-2021	Primero	Inglés 1	56
2020-2021	Primero	Educación Física 1	28
2020-2021	Primero	Historia de Cuba	52
2020-2021	Primero	Diseño y Programación Orientada...	80
2020-2021	Primero	Matemática Computacional	48
2020-2021	Primero	Economía Política	64
2020-2021	Primero	Matemática II	96
2020-2021	Primero	Diseño de Interfaces y Pruebas	32
2020-2021	Primero	Inglés 2	48

Figura 15: Pantalla de Asignaturas en Secretario Docente.

Informe de Base de Datos.docx - Microsoft Word

Conest INSERTAR DISEÑO DISEÑO DE PÁGINA REFERENCIAS CORRESPONDENCIA REVISAR VISTA

EVALUACIONES

Evaluaciones

Buscador

Curso Escolar: <Seleccionar un curso escolar> Año: <Seleccionar un año> Grupo Actual: <Seleccionar un grupo>

Asignatura: <Seleccionar una asignatura> Evaluación: <Seleccionar una calificación>

Listado de Evaluaciones

Curso Escolar ^	Año	Asignatura	Grupo Actual	Nombre y Apellidos	Calificación
2020-2021	Primero	Introducción a la Infor...	41	Nert Marchello	Aprobado
2020-2021	Primero	Introducción a la Pro...	41	Nert Marchello	Excelente
2020-2021	Primero	Matemática Discreta	41	Nert Marchello	Bien
2020-2021	Primero	Filosofía y Sociedad	41	Nert Marchello	Excelente
2020-2021	Primero	Matemática I	41	Nert Marchello	Bien
2020-2021	Primero	Inglés 1	41	Nert Marchello	Aprobado
2020-2021	Primero	Educación Física 1	41	Nert Marchello	Aprobado
2020-2021	Primero	Historia de Cuba	41	Nert Marchello	Bien
2020-2021	Primero	Diseño y Programaci...	41	Nert Marchello	Bien
2020-2021	Primero	Matemática Comput...	41	Nert Marchello	Aprobado
2020-2021	Primero	Economía Política	41	Nert Marchello	Aprobado
2020-2021	Primero	Matemática II	41	Nert Marchello	Excelente
2020-2021	Primero	Diseño de Interfaces	41	Nert Marchello	Aprobado

Figura 16: Pantalla de Evaluaciones en Secretario Docente.

Informe de Base de Datos.docx - Microsoft Word

Conest INSERTAR DISEÑO DISEÑO DE PÁGINA REFERENCIAS CORRESPONDENCIA REVISAR VISTA

ESCALAFÓN

Seleccionar alcance del Escalafón: ☒ Grupo ☐ Año

Año: Grupo:

Listado de Estudiantes

No.	No. Grupo	Nombre y Apellidos	Sexo	Municipio	Promedio Acumulado
1	8	Elbert Darree	Masculino	Plaza de la Revolución	4.33
2	25	Karrie Nern	Femenino	Arroyo Naranjo	4.33
3	2	Simone Bockh	Femenino	Boyeros	4.26
4	4	Cornie Bradfield	Masculino	Boyeros	4.26
5	7	Loella Coper	Femenino	Arroyo Naranjo	4.26
6	26	Saidee Nussey	Femenino	Arroyo Naranjo	4.26
7	14	Burton Hendrickson	Masculino	10 de Octubre	4.2
8	16	Jule Issacov	Masculino	Bauta	4.2
9	23	Gan McLeese	Masculino	Centro Habana	4.2
10	24	Karilynn Metheringham	Femenino	Centro Habana	4.2
11	28	Gonzalo Poynter	Masculino	Plaza de la Revolución	4.2
12	30	Artur Scarman	Masculino	Plaza de la Revolución	4.2
13	32	Kayley Trowbridge	Femenino	La Lisa	4.2
14	27	Turner Ortler	Masculino	Mayarí	4.13


Figura 17: Pantalla de Escalafón en Secretario Docente.

Informe de Base de Datos.docx - Microsoft Word


Conest INSERTAR DISEÑO DISEÑO DE PÁGINA REFERENCIAS CORRESPONDENCIA REVISAR VISTA

REPORTES


Selección de Reportes

Reporte 01



Listado de alumnos por grupo

Reporte 02



Listado de asignaturas de cada año en todos los cursos

Reporte 03



Listado de evaluaciones por curso escolar y por grupo en cada asignatura, pudiendo ser de un curso específico

Reporte 04



Listado los promedios anuales/finales obtenidos por los alumnos en cada grupo

Reporte 05



Escalafón anual o de grupo

Reporte 06



Certificación de notas de todos los estudiantes o de un estudiante específico

Reporte 07


Listado de alumnos desaprobados por grupos para un rango de cursos

Reporte 08


Listado histórico de los alumnos que causan baja por año/grupo

Reporte 09


Listado de repitentes por año

Figura 18: Pantalla de Selección de Reportes

3.4 Reportes diseñados:

Debido a los requisitos funcionales de la aplicación se diseñaron los siguientes reportes con ayuda del software JasperStudio:

- **Reporte 01 Listado de alumnos por grupo:** Permite mostrar un reporte con los diferentes alumnos agrupados por el curso escolar, año y grupo, mostrando para cada alumno (ordenados por su número de orden generado alfabéticamente por su apellido), el número de orden, nombre y apellidos y código del alumno.
- **Reporte 02 Listado de asignaturas que se imparten en cada año:** Permite mostrar un reporte con las diferentes asignaturas que se imparten, agrupadas por curso escolar y año, mostrando para cada asignatura su nombre y la cantidad de horas lectivas.
- **Reporte 03 Listado de las evaluaciones por grupo en cada asignatura:** Permite mostrar un reporte con las diferentes evaluaciones agrupadas por curso, año, grupo y asignatura, mostrando para cada alumno su número de orden, nombre y apellidos y evaluación obtenida en esa asignatura. Se puede restringir a un curso específico.
- **Reporte 04-A Listado de los promedios anuales obtenidos por los alumnos en cada grupo:** Permite mostrar un reporte con los promedios anuales de los alumnos, agrupándolos por año, curso escolar y el grupo actual, y mostrando para cada alumno el número de orden, nombre y apellidos, y el promedio anual.
- **Reporte 04-F Listado de los promedios finales obtenidos por los alumnos en cada grupo:** Permite mostrar un reporte con los promedios finales de los alumnos, agrupándolos por año y el grupo actual, y mostrando para cada alumno el número de orden, nombre y apellidos y el promedio acumulado.
- **Reporte 05-A Escalafón anual:** Permite mostrar un reporte con el escalafón anual de un año específico mostrando el curso escolar, el año, y para cada alumno, el número de escalafón, nombre y apellidos, promedio acumulado, número de orden, sexo y municipio de residencia. Esto es válido solo para el curso actual.
- **Reporte 05-G Escalafón de grupo:** Permite mostrar un reporte con el escalafón de un grupo específico mostrando el curso escolar, el año, el grupo, y para cada alumno, el número de escalafón, nombre y apellidos, promedio acumulado, número de orden, sexo y municipio de residencia. Esto es válido solo para el curso actual.
- **Reporte 06 Certificación de notas de un estudiante:** Permite mostrar un reporte la certificación de notas de un estudiante específico (o de todos los estudiantes), mostrando sus nombres y apellidos, y agrupando por cada año académico las evaluaciones de cada asignatura.

- **Reporte 07 Listado de alumnos desaprobados por grupos para un rango de cursos:** Permite mostrar un reporte con los alumnos desaprobados por grupo para un rango de cursos especificado, agrupándolos por curso, año y para cada estudiante, el grupo actual, el número de orden y las asignaturas desaprobadas.
- **Reporte 08-A Listado de los alumnos que causan baja de un año:** Permite mostrar un reporte con los alumnos que causan baja en un año específico, mostrando el curso, el año, y por cada alumno, el número de orden, su nombre y apellidos, el grupo y la causa de la baja. Este listado es histórico, y muestra todos los alumnos baja almacenados en el sistema, que han sido reportados baja en ese año (primero, segundo, tercero o cuarto).
- **Reporte 08-G Listado de los alumnos por grupo que causan baja de un año:** Permite mostrar un reporte con los alumnos que causan baja por grupo del curso actual de un año específico, mostrando agrupados el curso escolar, año y grupo, y para cada alumno el número de orden, su nombre y apellidos, y la causa de la baja. Este listado es histórico, y muestra todos los alumnos baja almacenados en el sistema que han sido reportados baja en ese año (primero, segundo, tercero o cuarto).
- **Reporte 09 Listado de repitentes por año:** Permite mostrar un reporte con los alumnos repitentes agrupados por curso (actual) y año, y por cada alumno el número de orden, los nombres y apellidos y el grupo.

Conclusiones

Las principales conclusiones se encuentran a continuación:

- De manera general se pueden considerar por cumplidos todos los objetivos trazados inicialmente.
- El sistema automatizado desarrollado cumple íntegramente con los requisitos funcionales identificados.
- Durante el proceso de desarrollo se han identificado otros posibles requisitos funcionales, los cuales serán agregados en el apartado siguiente a modo de recomendaciones para futuras versiones del software.
- La mayoría de las decisiones de diseño o implementación tomadas antes o durante el desarrollo han sido documentadas en este informe y se ha explicado detalladamente el porqué de las mismas.
- En este punto se da por cumplida la premisa fundamental del proyecto, crear un sistema automatizado para el control docente de una institución.

Recomendaciones

A continuación, se describen una lista de funcionalidades que podrían implementarse en el futuro o mejoras que incrementarían el valor de las ya existentes, así como otras consideraciones que se exponen:

- Agregar las entidades curso escolar y plan escolar, lo cual facilite el trabajo con estas entidades, y permita agregar funcionalidades en el futuro.
- Crear una solución web con este sistema para así facilitar el acceso y uso de la aplicación.
- Almacenar la base de datos del sistema en un servidor propio, o empleando un servicio de hosting externo como AWS Services.
- Optimizar algunos detalles de la interfaz para que la misma sea más amigable e intuitiva para el usuario, así como robusta y consistente con mayor número de funcionalidades.
- Implementar un módulo de la aplicación orientado a calcular y mostrar las métricas del negocio. Un subsistema encargado de generar gráficos y resúmenes a partir de la información de la Base de Datos, que no sean los reportes ya generados en la aplicación.

Referencias Bibliográficas

- [1] C. J. Date., Introducción a los Sistemas de Bases de Datos, 7ma edición: Prentice Hall.
- [2] R. M. M. García, Sistemas de Bases de Datos, La Habana: Editorial Félix Varela, 2005.
- [3] Manual de usuario de ERECASE 2.1.
- [4] M. C. pgadmin3.chm, Ayuda Postgres.
- [5] M. c. d. P. y. d. PL/pgSQL.
- [6] A. S. y. Y. Vazquez, PL/pgSQL y otros lenguajes procedurales en PostgreSQL, La Habana: Ediciones Futuro, 2016.
- [7] E. & Navathe, Chapter 2: Database Systems Concepts and Architecture in Fundamentals of Database Systems, Addison Wesley, 2005.
- [8] H. H. T. y. S. G. Chase, The software vulnerability guide programming, Massachusetts: Charles River Media, Inc. Hingham, 2005.

Anexos

Anexo 1: Script de creación de las tablas de la base de datos

CREATE TABLE STUDENT (

```
id_std SERIAL PRIMARY KEY NOT NULL,  
names_std VARCHAR(256) NOT NULL  
CHECK(CHARACTER_LENGTH(TRIM(names_std)) > 0),  
sex_std CHAR NOT NULL CHECK(sex_std='F' OR sex_std='M'),  
mun_std VARCHAR(64) NOT NULL  
CHECK(CHARACTER_LENGTH(TRIM(mun_std)) > 0),  
id_y_gr_std INTEGER NOT NULL,  
lastnames_std VARCHAR(256) NOT NULL  
CHECK(CHARACTER_LENGTH(TRIM(lastnames_std)) > 0),  
type_std CHAR NOT NULL CHECK(type_std='P' OR type_std='R' OR  
type_std='D'),  
id_gr_std INTEGER NOT NULL);
```

CREATE TABLE REPEATER (

```
id_std INTEGER PRIMARY KEY NOT NULL);
```

CREATE TABLE PROMOTED (

```
id_std INTEGER PRIMARY KEY NOT NULL);
```

CREATE TABLE CAUSE_OF_DISMISSAL (

```
id_dm SERIAL PRIMARY KEY NOT NULL,  
dm VARCHAR(128) NOT NULL UNIQUE  
CHECK(CHARACTER_LENGTH(TRIM(dm)) > 0));
```

CREATE TABLE DISMISSAL (

```
id_std INTEGER PRIMARY KEY NOT NULL,  
id_dm_dm INTEGER NOT NULL);
```

CREATE TABLE ACADEMIC_YEAR (

```
id_y SERIAL PRIMARY KEY NOT NULL,  
year_y INTEGER NOT NULL CHECK(year_y BETWEEN 1 AND 7),
```

school_course_start_y INTEGER NOT NULL CHECK(*school_course_start_y* BETWEEN 2000 AND 2100),

school_course_end_y INTEGER NOT NULL CHECK(*school_course_end_y* BETWEEN 2000 AND 2100));

CREATE TABLE GROUP_ST (

id_gr SERIAL NOT NULL,

id_y_gr INTEGER NOT NULL,

gr_number INTEGER NOT NULL CHECK(*gr_number*>0),

classroom_gr INTEGER NOT NULL CHECK(*classroom_gr* >= 0),

PRIMARY KEY (*id_gr*,*id_y_gr*));

CREATE TABLE ACADEMIC_SUBJECT (

id_sub SERIAL PRIMARY KEY NOT NULL,

name_sub VARCHAR(128) NOT NULL UNIQUE
CHECK(CHARACTER_LENGTH(TRIM(*name_sub*)) > 0),

hours_plan_sub INTEGER NOT NULL CHECK(*hours_plan_sub* > 0),

id_y_sub INTEGER NOT NULL);

CREATE TABLE TYPE_OF_EVALUATION (

id_eval SERIAL PRIMARY KEY NOT NULL,

eval VARCHAR(60) NOT NULL UNIQUE
CHECK(CHARACTER_LENGTH(TRIM(*eval*)) > 0),

eval_num INTEGER NOT NULL UNIQUE CHECK(*eval_num* >= 0));

CREATE TABLE EVALUATION (

id_sub_ev INTEGER NOT NULL,

id_std_ev INTEGER NOT NULL,

id_eval_ev INTEGER,

PRIMARY KEY (*id_sub_ev*,*id_std_ev*));

CREATE TABLE USER_APP (

id_user SERIAL PRIMARY KEY NOT NULL,

id_role_user INTEGER NOT NULL,
mail_user VARCHAR(128) NOT NULL UNIQUE
CHECK(CHARACTER_LENGTH(TRIM(*mail_user*)) > 0),
password_user VARCHAR(64) NOT NULL
CHECK(CHARACTER_LENGTH(*password_user*) > 0),
id_std_user INTEGER);

CREATE TABLE ROLE_USER(

id_role SERIAL PRIMARY KEY NOT NULL,
type_role VARCHAR(64) NOT NULL UNIQUE
CHECK(CHARACTER_LENGTH(TRIM(*type_role*)) > 0));

ALTER TABLE STUDENT ADD CONSTRAINT

STUDENT_foreign_key_GROUP_ST

FOREIGN KEY (*id_y_gr_std*, *id_gr_std*) REFERENCES *GROUP_ST*(*id_y_gr*,
id_gr) ON DELETE CASCADE;

ALTER TABLE REPEATER ADD CONSTRAINT

REPEATER_id_std_STUDENT_id_std

FOREIGN KEY (*id_std*) REFERENCES *STUDENT*(*id_std*) ON DELETE
CASCADE;

ALTER TABLE PROMOTED ADD CONSTRAINT

PROMOTED_id_std_STUDENT_id_std

FOREIGN KEY (*id_std*) REFERENCES *STUDENT*(*id_std*) ON DELETE
CASCADE;

ALTER TABLE DISMISSAL ADD CONSTRAINT

DISMISSAL_id_std_STUDENT_id_std

FOREIGN KEY (*id_std*) REFERENCES *STUDENT*(*id_std*) ON DELETE
CASCADE;

ALTER TABLE DISMISSAL ADD CONSTRAINT

DISMISSAL_id_dm_dm_CAUSE_OF_DISMISSAL_id_dm

FOREIGN KEY (*id_dm_dm*) REFERENCES *CAUSE_OF_DISMISSAL*(*id_dm*)
ON DELETE CASCADE;

ALTER TABLE GROUP_ST ADD CONSTRAINT

GROUP_ST_id_y_gr_ACADEMIC_YEAR_id_y

*FOREIGN KEY (id_y_gr) REFERENCES ACADEMIC_YEAR(id_y) ON
DELETE CASCADE;*

ALTER TABLE ACADEMIC_SUBJECT ADD CONSTRAINT

ACADEMIC_SUBJECT_id_y_sub_ACADEMIC_YEAR_id_y

*FOREIGN KEY (id_y_sub) REFERENCES ACADEMIC_YEAR(id_y) ON
DELETE CASCADE;*

ALTER TABLE EVALUATION ADD CONSTRAINT

EVALUATION_id_sub_ev_ACADEMIC_SUBJECT_id_sub

*FOREIGN KEY (id_sub_ev) REFERENCES ACADEMIC_SUBJECT(id_sub)
ON DELETE CASCADE;*

ALTER TABLE EVALUATION ADD CONSTRAINT

EVALUATION_id_std_ev_STUDENT_id_std

*FOREIGN KEY (id_std_ev) REFERENCES STUDENT(id_std) ON DELETE
CASCADE;*

ALTER TABLE EVALUATION ADD CONSTRAINT

EVALUATION_id_eval_ev_TYPE_OF_EVALUATION_id_eval

*FOREIGN KEY (id_eval_ev) REFERENCES TYPE_OF_EVALUATION(id_eval)
ON DELETE CASCADE;*

ALTER TABLE USER_APP ADD CONSTRAINT

USER_APP_id_role_user_ROLE_USER_id_role

*FOREIGN KEY(id_role_user) REFERENCES ROLE_USER(id_role) ON
DELETE CASCADE;*

ALTER TABLE USER_APP ADD CONSTRAINT

USER_APP_id_std_user_ROLE_USER_id_std

*FOREIGN KEY(id_std_user) REFERENCES STUDENT(id_std) ON DELETE
CASCADE;*