

Escuela Politécnica Nacional

Programación II

2024-A



Proyecto final

Tema:

Sistema de ingreso estudiantil

Grupo 6

Profesor:

Ing. Paccha Angamarca Patricio Michael

Integrantes:

Rios Condoy Marco David

Rosas Borja Ariel Alexander

Sacoto Solorzano Jonathan David

Sanchez Vistin Katherine Abigail

Quito, 2024

Estándares de codificación
Sistema de ingreso estudiantil
2024-A

Contenido

1.	Marco general	4
1.1.	Generalidades	4
2.	Nomenclatura.....	4
2.1.	Regla general	4
2.2.	Reglas específicas.....	5
2.2.1.	Instancia.....	5
2.2.2.	Base de datos	5
2.2.3.	Esquema	5
2.2.4.	Columnas.....	6
2.2.5.	PRIMARY KEY CONSTRAINT	6
3.	Identificadores	6
3.1.	Parámetros	6
3.2.	Variables	7
4.	Documentación.....	7
4.1.	Encabezado	7
4.2.	Nomenclatura de Scripts de Bases de Datos	8
4.3.	Número de Script y Fechas.....	8
4.4.	Documentación de Objetos y Generación de Diccionario de Datos	8
4.4.1.	Declaración de Variables de Trabajo:	8
4.4.2.	Documentar una Tabla:	9
5.	Buenas prácticas	9
5.1.	Variables columnas y campos usar tipos Unicode.....	9
5.2.	Palabras reservadas.....	9
5.3.	Sangría.....	9
5.4.	Uso de espacios en blanco	9
5.5.	Comentarios de código	9
5.6.	Diagrama de entidad relación y diccionario de datos.....	10
5.7.	Optimización de consultas.....	10
5.8.	Store Procedures	10
6.	Recomendaciones	10
6.1.	Uso de Comentarios Claros:	10
6.2.	Validaciones y Chequeos Antes de Cambios	11
6.3.	Transacciones y Control de Errores:.....	11
6.4.	Manejo de Datos en Inserciones y Actualizaciones:	11

6.5. Verificación de la Integridad de Datos:	11
7. Referencias	11

1. Marco general

La optimización del sistema de acceso estudiantil se realizará mediante la implementación de códigos QR, lo que permitirá una validación eficiente de la pertenencia de los estudiantes registrados en la Escuela Politécnica Nacional (EPN). Este sistema estará integrado con las bases de datos existentes, incluyendo el Sistema de Información Institucional (SII) y el Sistema de Administración de Estudiantes y Workflows (SAEW), que forman parte de la infraestructura de la Escuela Politécnica Nacional. Además, se desarrollará una interfaz gráfica que permitirá al personal administrativo realizar cambios en la base de datos de manera intuitiva y eficiente.

1.1.Generalidades

- Normaliza las tablas para asegurar que no exista duplicidad de datos y se aproveche al máximo el almacenamiento en las tablas.
- Mostrar un modelo de clases que representan los datos de los estudiantes y sus estados de acceso.
- Vista de las clases responsables de la interfaz de usuario, incluyendo la pantalla de escaneo y la pantalla de la base de datos de estudiantes.
- Controlador de las clases que manejan la lógica de negocio, como la lectura y procesamiento de códigos QR.
- Utilidades de las clases utilitarias que proporcionan funciones auxiliares, como la validación de códigos QR.

2. Nomenclatura

2.1.Regla general

- Usar nombres en singular para todos los objetos a crear.
- No usar palabras reservadas en los nombres de los objetos, en el caso de ser absolutamente necesario usar [], por ejemplo [Year].
- Los nombres de los objetos de base de datos utilizarán el < sufijo> (en mayúsculas) y estarán compuestos por palabras con la primera letra en mayúscula (UpperCamelCase). Se debe evitar el uso de abreviaciones que dificulten la comprensión.
- Si se requiere separar el nombrado de los objetos debe usar el underscore: “ _ ”

2.2.Reglas específicas

2.2.1. Instancia

Reglas:

- Las instancias deben ser descriptivas del propósito del sistema.
- Usa mayúsculas y evita espacios en los nombres.
- Usa un formato consistente para todas las instancias relacionadas.

Formato:

<APLICATIVO>_<DESCRIPCIÓN>

Ejemplo:

QRSTUDENT

Instancia principal para la validación de acceso mediante QR para estudiantes.

2.2.2. Base de datos

Reglas:

- El nombre de la base de datos debe reflejar claramente el nombre del aplicativo.
- El nombre físico de la base de datos -Master Database Files y Database Files- debe corresponder al alias de la aplicación seguido de guion bajo más las palabras Data.

Formato:

BDD:<APLICATIVO>

MDF: <ALIAS>_Data

LDF: <ALIAS>_Log

Ejemplo:

BDD: QR_STUDENT_VALIDATOR

MDF: QRS_Data

LDF: QRS_Log

2.2.3. Esquema

Reglas:

- Nombre en singular y UpperCammelCase. Si la tabla es relacional considere que primero es el nombre de la tabla principal.

Formato:

<TablaNombre>

Ejemplo:

Estudiantes

2.2.4. Columnas

Reglas:

- Para las claves primarias se debe usar el prefijo Id unido al <TablaNombre>
- No sobrecargar la descripción.
- Las columnas que sirven como referencia como un Foreign Key deberán ser nombradas como esta en la Tabla que hace referencia.

Formato:

<ColumNombre>

Ejemplo:

IdEstudiante → PRIMARY KEY

2.2.5. PRIMARY KEY CONSTRAINT

Reglas:

- Usar el id nombre de la columna seguido del PRIMARY KEY constraint.

Formato:

<NombreColumna> PRIMARY KEY constraint

Ejemplo:

IdEstudiante PRIMARY KEY constraint.

3. Identificadores

Reglas:

- LowerCamelCase: Todos los identificadores deben seguir la convención LowerCamelCase.

Formato:

<nombreParámetro>

Ejemplo:

Variable:

numeroAutorizacion

Constante:

maximoIntentos

3.1. Parámetros

Reglas:

- Los parámetros deben llevar un prefijo seguido de UpperCamelCase para mayor claridad.

Prefijos:

Entrada (Input): `i`

Salida (Output): `o`

Entrada/Salida (Input Output): `io`

Formato:

<tipodato><NombreParámetro>

Ejemplo:

```
CREATE PROCEDURE PA_EEL_ResortearFirmante (
    @iCedula VARCHAR(10) INPUT,
    @oSecretario INT OUTPUT
)
AS
BEGIN
    -- Declaración de variables
    -- Sentencias SQL
END
```

3.2.Variables

Reglas:

- Las variables deben seguir la convención LowerCamelCase.

Formato:

<nombreParámetro>

Ejemplo:

```
DECLARE numeroAutorizacion INT;
```

4. Documentación

4.1.Encabezado

Cada script, procedimiento y función en tu proyecto debe tener un encabezado que incluya detalles del autor, descripción, y validación. Aquí tienes un ejemplo adaptado:

Ejemplo:

```
/*-----\
```

| ©Copyright

| Todos los Derechos Reservados

| Author: Katherine Sanchez

| Description : Procedimiento para validar el acceso de estudiantes mediante códigos QR.

| Fecha : 2024-08-08

|-----|

4.2.Nomenclatura de Scripts de Bases de Datos

Reglas:

Los scripts de modificación de vistas, procedimientos y funciones deben seguir la nomenclatura:

Formato:

CJ-DNTICS-SDSW-<DATABASE>-<SHEMMA>-
<Vista,función,procedimiento>.sql

Ejemplo:

CJ-DNTICS-SDSW-QRCodeStudentValidator-CLEX1-vs_ValidarAcceso.sql

4.3.Número de Script y Fechas

Mantener la secuencia del número de script.

Las fechas son relevantes para el seguimiento.

4.4.Documentación de Objetos y Generación de Diccionario de Datos

4.4.1. Declaración de Variables de Trabajo:

DECLARE @esquema NVARCHAR(50);

DECLARE @tabla NVARCHAR(100);

DECLARE @comentario NVARCHAR(2000);

DECLARE @columna NVARCHAR(100);

4.4.2. Documentar una Tabla:

Ejemplo:

SET @esquema = 'clex1';

SET @tabla = 'Estudiantes';

SET @comentario = 'Tabla que almacena la información de los estudiantes de la universidad.';

5. Buenas prácticas

5.1. Variables columnas y campos usar tipos Unicode.

Usar tipos como: nchar, nvarchar y ntext

5.2. Palabras reservadas

Todas las palabras reservadas deberán escritas ser en mayúsculas.

5.3. Sangría

- Las instrucciones serán alineadas de acuerdo con la profundidad los diferentes bloques que se vayan construyendo.
- Se puede utilizar alguna herramienta para dar formato al código.

5.4. Uso de espacios en blanco

Siempre incluya un espacio entre cada identificador y separador.

Ejemplo:

En lugar de esto:

WHILE (totalSales < maximumSales AND companyType='NEW') LOOP

Escriba esto:

WHILE (totalSales < maximumSales AND companType = 'NEW') LOOP

5.5. Comentarios de código

- Se debe tomar en cuenta las siguientes consideraciones para comentar el código:
- No se debe comentar cada línea del código es decir solo llevar una documentación adecuada.
- Comentar solamente las partes importantes del código.

5.6.Diagrama de entidad relación y diccionario de datos

Generar para la Base de datos (BDD) un Diagrama Entidad Relación (ER) y un Diccionario de Datos Script que crean objetos de base de datos deben ser documentados.

5.7. Optimización de consultas

- Se procurará elegir en la cláusula WHERE aquellos campos que formen parte de la clave primaria.
- Si se desea consultar por campos pertenecientes a índices compuestos es mejor utilizar todos los campos que componen el índice.
- Cuando se utilizan varias tablas dentro de la consulta hay que tener cuidado con el orden empleado en la cláusula FROM. Si se desea saber cuántos alumnos se matricularon en el año 1996 se escribe de la siguiente manera:

```
FROM Alumnos, Matriculas
```

```
WHERE Alumno.IdAlumno = Matriculas.IdAlumno AND Matriculas.Año = 1996
```

El gestor recorrerá todos los alumnos para buscar sus matriculas y devolver las correspondientes. Si se escribe:

```
FROM Matriculas, Alumnos
```

```
WHERE Matriculas.Año = 1996 AND Matriculas.IdAlumno = Alumnos.IdAlumnos
```

El gestor filtra las matrículas y después selecciona los alumnos, de esta forma tiene que recorrer menos registros.

5.8.Store Procedures

En la programación de Store Procedures se debe utilizar mayúsculas para las sentencias propias y palabras reservadas del SQL.

6. Recomendaciones

6.1.Uso de Comentarios Claros:

- Usar comentarios claros y concisos para explicar partes complejas del script. Esto facilita la comprensión del código y su mantenimiento por parte de otros desarrolladores o administradores de bases de datos.

6.2. Validaciones y Chequeos Antes de Cambios

- Antes de crear o modificar objetos, se recomienda realizar validaciones para asegurarse de que los objetos no existen o no se modifican de manera incorrecta.

6.3. Transacciones y Control de Errores:

- Asegurarse de que las operaciones críticas (como cambios de estructura y datos) estén envueltas en transacciones para garantizar la atomicidad. Usa bloques Try -catch para manejar errores.

6.4. Manejo de Datos en Inserciones y Actualizaciones:

- Al insertar o actualizar datos, se tiene que especificar las columnas para evitar problemas de mapeo y asegurar la integridad de los datos. Asegúrate de que las columnas especificadas coincidan con las esperadas en la tabla.

6.5. Verificación de la Integridad de Datos:

- Asegurarse de que las operaciones de modificación o eliminación afecten sólo a los registros correctos, preferiblemente utilizando la clave primaria (PK) o criterios específicos para evitar cambios no deseados.

7. Referencias

Consejo de la Judicatura del Ecuador. (2018). *Estándares de desarrollo de base de datos de la DNTICS (Versión 3.0)*. Dirección Nacional de Tecnologías de la Información y Comunicaciones.