***Logotipo, nombre de la empresa

Descripción generada automáticamente***

***Instituto Tecnológico de Costa Rica***

***Centro Académico de Limón***

***Curso:***

***Bases de Datos I***

***Grupo: 60***

***Proyecto01***

***Profesor:***

***Allan Rodríguez Dávila***

***Estudiante:***

***Jesús Cruz López (2020065267)***

***II Semestre - 2021***

***Fecha de entrega: 05/10/2021***

Índice

[**Manual de usuario** 1](#_Toc84286381)

[**Pruebas de funcionalidad** 2](#_Toc84286382)

[**Descripción del problema** 3](#_Toc84286383)

[**Diseño del programa** 5](#_Toc84286384)

[**Diagrama Entidad – Relación** 6](#_Toc84286385)

[**Diagrama Relacional** 7](#_Toc84286386)

[**Librerías utilizadas** 8](#_Toc84286387)

[**Análisis de resultados** 8](#_Toc84286388)

[**Bitácora** 8](#_Toc84286389)

# **Manual de usuario**

# **Pruebas de funcionalidad**

# **Descripción del problema**

El sistema deberá disponer de un módulo para manejo de usuarios, donde se podrá crear usuarios (profesores, estudiantes y padres), modificar la información de los usuarios, crear roles de usuarios, asignación de roles a usuarios, definir la visibilidad y acceso de las opciones y funcionalidades del sistema por roles de usuarios (por módulo tres opciones: edición, visualización y reportería). Por cada usuario se debe almacenar la siguiente información: cédula, nombre completo, sexo, fecha nacimiento, edad (calculada), provincia, lugar de residencia, teléfono, fecha de creación. Debe manejar tres roles por defecto:

• Profesor: se pueden asignarle cursos.

• Estudiante: se le matricula en cursos.

• Padre: se le asocian hijos.

El sistema debe disponer de un sistema de gestión de profesores, que permita registrar, editar, eliminar (validar restricciones), visualizar y generar reportes de la información de los profesores. Estos deben ser usuario con el rol de profesor y adicionalmente se les incluye un salario, una materia que imparten. A estos se les pueden asignar grupos y además manejar un histórico de salario (inicio, fin, monto). Además, hará uso de la gestión de evaluaciones. El sistema manejará reportes, como, por ejemplo: horario por período, histórico de salarios, cursos por profesor, rendimiento estudiantil; aplicando filtros a la información, como por ejemplo rango de fechas, entre otros.

El sistema debe disponer de un sistema de gestión de padres, que permita registrar, editar, eliminar (validar restricciones), visualizar y generar reportes de la información de los padres. Estos deben ser usuario con el rol de padre y adicionalmente se les incluye una profesión u oficio y conyugue, teléfono de conyugue. Adicionalmente, se debe llevar control de los cobros adeudados por costos de matrícula y mensualidad de sus hijos y los pagos realizados. Además, hará uso de la gestión de evaluaciones para visualizar información de los hijos. Debe permitir hacer pago de matrícula y mensualidades de los hijos seleccionando algunos de los recibos pendientes generados por el sistema en el momento de la matrícula. El sistema manejará reportes, como, por ejemplo: hijos por padre, histórico de pagos, saldos, entre otros.

El sistema debe disponer de un sistema de gestión de estudiantes, que permita registrar, editar, eliminar (validar restricciones), visualizar y generar reportes de la información de los estudiantes. Estos deben ser usuario con el rol de estudiantes y adicionalmente se les asigna un padre (del sistema) y guarda el grado que cursará, guarda un histórico de grados cursados, indicando el período y el estado en ese período. A estos se les pueden matricular en un grado lectivo y automáticamente se les asignarán los grupos de ese grado en el período indicado. Además, hará uso de la gestión de evaluaciones (visualiza) donde podrá ver por cada período lectivo su horario de clases (curso, aula, profesor y horario), y la matriz de evaluaciones para cada curso. El sistema manejará reportes, como, por ejemplo: histórico de grupos, informe de evaluaciones por período, cursos matriculados, cursos perdidos, cursos aprobados, entre otros.

El sistema deberá contener un módulo para gestionar los períodos lectivos, permitiendo crear, modificar, eliminar y visualizar la información de los períodos. Por cada período se guarda el año, número de período, fecha inicio, fecha final. Un período abierto se puede cerrar (sólo se permite un período abierto), esto implicará que para los grupos de ese período se les hará un cierre, generando información por cada estudiante donde se indica si está aprobado o reprobado, según la nota almacenada y la nota mínima del período asociada al curso. Debe informar los grupos que no se pudieron cerrar y brindar una opción de cerrar grupos pendientes. Una vez que todos los grupos estén cerrados para el período se debe analizar el estado de aprobación de los estudiantes y manejar el reporte del período y el historial y actualizar el grado del estudiante en caso de que haya aprobado todos los grupos en los que esté matriculado. Adicionalmente se dará mantenimiento a los grados que se imparten ese período, los grados son secuenciales, empezando en 1.

El sistema deberá contener un módulo para gestionar los grupos, permitiendo crear, modificar, eliminar y visualizar la información de los grupos. Para cada grupo se guarda el código de grupo, nombre, profesor a cargo, el período, cupo, materia y el grado al que pertenece. Cada grupo se le permite asociar estudiantes matriculados (automáticamente con la matrícula) y el profesor tendrá acceso a la información general del curso. Además. el profesor tendrá las siguientes funcionalidades:

• Gestión de evaluaciones: deben sumar 100 y permite dar mantenimiento a las evaluaciones que tendrá el grupo, asignar notas por estudiante por evaluación. Además, ver reporte de notas y aprobados y reprobados.

• Estudiantes: podrá ver la lista de estudiantes y llevar un control de asistencia, permite registrar asistencia por día y ver reportes de asistencia.

Cuando se cierra un período se deben cerrar todos los grupos de ese período que posean su matriz de evaluaciones completa. El sistema deberá contener un módulo para gestionar la matrícula de estudiantes incluir, borrar y visualizar la información de matrícula. Para matrícula se debe indicar la cédula del estudiante y el período a matricular, el sistema automáticamente verificará el grado correspondiente a matricular y lo incluirá en el primer grupo, de cada materia, del grado correspondiente con espacio disponible. Adicionalmente, se le generarán los cobros pendientes al padre del estudiante en el módulo de pagos, todos quedarán pendientes: se le genera un cobro por el monto de matrícula (debe tener un detalle el cobro indicando período, grado y estudiante) y doce cobros por cada grupo, por el monto de curso, guardando la fecha de creación, el usuario, por cada grupo matriculado (debe tener un detalle del cobro que indique período, grado, grupo, consecutivo de 1 a 12 y estudiante). Toda la información de matrícula debe poder consultarse por estudiante (período, grado, grupos, cobros).

El sistema deberá contener un módulo para gestionar la gestión de pagos de padres de familia, permitir incluir, borrar y visualizar la información de pagos y cobros. Por cédula de padre se puede consultar su histórico de pago, detallado por cada uno (incluyendo toda la información y los detalles de la factura) e indicando total. Además, ver cobros pendientes y seleccionarlos y facturarlos, se pueden seleccionar varios cobros (deben ordenarse por estudiante, período, y consecutivo) y facturarlos, la factura debe asociarse a los cobros y guardar fecha de pago, usuario, monto total, se calcula impuesto del 2%, se debe mostrar el detalle de los pagado. Las facturas se pueden consultar y filtrar por estudiantes, padre y/o por fecha. Además, se debe dar mantenimiento a los costos por matrícula (por grado) y mensualidad por grupo por período.

La aplicación debe maximizar el uso de campos catálogos para evitar que el usuario disponga para escribir en muchos campos libres. Todos aquellos campos que se identifiquen tipo catálogo se les debe generar una tabla. La parte aplicativa deberá tener un diseño elegante y amigable con el usuario.

# **Diseño del programa**

## **Diagrama Entidad – Relación**

Diagrama, Esquemático

Descripción generada automáticamente

## **Diagrama Relacional**

# **Librerías utilizadas**

# **Análisis de resultados**

|  |  |  |  |
| --- | --- | --- | --- |
| Requerimiento | Logrado | No Logrado |  |
| Diagrama Entidad - Relación | x |  |  |
| Diagrama Relacional | x |  |  |
| Diccionario |  |  |  |
| Diseño de la base de datos(script) | x |  |  |
| Back - End |  |  |  |
| Front - End |  |  |  |
| Funcionalidad (Agregar, eliminar y modificar) |  |  |  |

# **Bitácora**

Repositorio de Github: <https://github.com/jesus-cl27/Proyecto01-BasesDatos-I>