**Instituto Nacional de Aprendizaje**

**Programador de Aplicaciones Informáticas**

**Módulo Gestión de Base de Datos**

**Proyecto Individual**

**Fase II**

**Estudiante: Mariel Daniela Rojas Sanchez**

**Cedula: 208030487**

**Profesor: Luis Alonso Bogantes**

**06 febrero 2022**

Índice

[Introducción: 3](#_Toc126570804)

[Objetivo General: 3](#_Toc126570805)

[Objetivos Específicos: 4](#_Toc126570806)

[Definición del problema. 4](#_Toc126570807)

[Requerimientos funcionales 6](#_Toc126570808)

[Reglas de negocio 8](#_Toc126570809)

[Modelo Conceptual 10](#_Toc126570810)

[Planteamiento conceptual de la solución. 11](#_Toc126570811)

[Modelo lógico 13](#_Toc126570812)

[Diccionario de Datos 13](#_Toc126570813)

[**Matrículas** 13](#_Toc126570814)

[**Estudiantes** 14](#_Toc126570815)

[**Detalle Matriculas** 14](#_Toc126570816)

[**Módulos\_Disponibles** 15](#_Toc126570817)

[**Horario** 15](#_Toc126570818)

[**Profesores** 15](#_Toc126570819)

[**Módulo** 16](#_Toc126570820)

[**Programa Módulo** 16](#_Toc126570821)

[**Programas** 17](#_Toc126570822)

[**Laboratorios** 17](#_Toc126570823)

[**Planillas** 17](#_Toc126570824)

[Conclusiones: 18](#_Toc126570825)

# Introducción:

Este proyecto tiene como fin poner en práctica todos los temas vistos a lo largo del módulo gestión de base de datos; En este módulo se aprende sobre el diseño de bases de datos, la implementación de base de datos relacionales y sus componentes según las normas establecidas y requerimientos de la clientela, así como también el generar consultas en lenguaje TSQL.

En este proyecto se va a analizar detalladamente y desarrollar una base de datos encargada de realizar el proceso de matrícula de una universidad, por lo que principalmente se requiere que la Base de Datos administre la información de los estudiantes, la matrícula, el pago de los módulos, las calificaciones obtenidas en cada materia y la información de los docentes. Además, debe administrar la información de los programas que se imparten en la universidad y los módulos comprendidos en cada uno de los programas.

Para realizar la implementación de la base de datos es necesario hacer un análisis previo, donde se determine detalladamente el problema que debe de solucionar la base de datos, para ello es indispensable que se determinen los requerimientos funcionales, las reglas de negocio, también es necesario plantear a nivel conceptual la solución y crear el modelo lógico de datos con su respectivo diccionario de datos.

# Objetivo General:

Implementar y diseñar una base de datos que sea capaz de administrar toda la información de los estudiantes, profesores y laboratorios de una Universidad para que estos puedan hacer efectiva su matrícula.

# Objetivos Específicos:

* Realizar un análisis profundo del problema que debe solucionar la base de datos.
* Determinar los requerimientos funcionales y las reglas básicas del negocio.
* Crear el modelo conceptual de la base de datos.
* Plantear a nivel conceptual la solución.
* Crear el modelo lógico de la base de datos.
* Crear el diccionario de datos de la base de datos.

# Definición del problema.

La Universidad **“Fuente del Conocimiento”** va a brindar sus servicios de enseñanza a partir el año 2026 de forma presencial, y entre sus prioridades está el diseño de la Base de Datos.

Se requiere que la Base de Datos administre la información de los estudiantes, la matrícula, el pago de los módulos, las calificaciones obtenidas en cada materia y la información de los docentes. Además, debe administrar la información de los programas que se imparten en la universidad y los módulos comprendidos en cada uno de los programas.

Usted ha sido contratado para el diseño de la base de datos en 3 etapas: (1) Diseño, (2) Implementación y (3) la creación de los scripts de consultas, procedimientos almacenados, funciones, etc.

Es importante que contemple la siguiente información en el diseño, análisis e implementación del sistema de Bases de Datos que se necesita.

1. La Universidad **“Fuente del Conocimiento”** va a impartir 3 programas, los cuales se mencionan a continuación:
   1. **Programación de Dispositivos Móviles.**
   2. **Programación de Sistemas de Escritorio**
   3. **Programación de Páginas Web**
2. El programa “**Programación de dispositivos móviles**”, está conformado por cuatro módulos.
   1. **Programación para Android I**, duración: 120 horas, costo: ₡250 000.
   2. **Programación para Android II**, duración: 110 horas, costo: ₡240 000.
   3. **Programación para Android III**, duración: 100 horas, costo: ₡220 000.
   4. **Programación para Aplicaciones Mixtas**, duración 100 horas, costo: ₡195 000.
3. El programa “**Programación de sistemas de escritorio**”, está conformado por tres módulos.
   1. **Lógica Computacional**, duración: 140 horas, costo: ₡225 000.
   2. **Introducción a Java**, duración: 150 horas, costo: ₡335 000.
   3. **Programación Orientada a Objetos**, duración: 160 horas, costo: ₡345 000.
4. El programa “**Programación de Páginas Web**”, está conformado por cinco módulos.
   1. **HTML**, duración: 90 horas, costo: ₡185 000.
   2. **CSS**, duración: 60 horas, costo: ₡175 000.
   3. **JavaScript**, duración: 70 horas, costo: ₡180 000.
   4. **Bootstrap**, duración: 90 horas, costo: ₡185 000.
   5. **Nodo JS**, duración: 90 horas, costo: ₡185 000.
5. El campus universitario cuenta con excelentes instalaciones y cuenta con 5 laboratorios equipados para impartir las clases.
6. El horario de la universidad va a ser de 7:00am a 6:00pm de lunes a viernes, por lo tanto, las lecciones deben estar contempladas en ese horario.
7. Los módulos pueden tener un mínimo de dos horas y un máximo de 5 horas diarias, pero en un único bloque al día, es decir, un módulo no se puede impartir en la mañana y luego en la tarde del mismo día. Además, se puede impartir: uno, dos, tres, cuatro o cinco días por semana.
8. Cada módulo puede tener más de un grupo abierto con su respectivo horario.
9. Cuando un/a estudiante vaya a realizar la matrícula de un módulo, debe indicar el módulo y el grupo en el que desea matricularse.
10. Cuando se abre un módulo se solicita al usuario la fecha de inicio. El sistema calculará la fecha final considerando los datos anteriores (días y horarios en que se impartirá), además de los días feriados, los cuales no son lectivos.
11. El sistema debe registrar los pagos realizados por los estudiantes.
12. Los profesores deben tener una certificación para impartir un determinado módulo. Un profesor puede tener certificaciones para impartir diferentes módulos, mientras que otro sólo podría tener una sola certificación, por lo que sólo podrá impartir un solo módulo en la Universidad.
13. Cada módulo tiene calificaciones finales, cuya nota mínima es de 70 para continuar en el programa. Debe verificar el cumplimiento de este requisito en cada nueva matrícula.
14. Un estudiante no va a recibir la nota final si no ha cancelado el módulo, por ende, tampoco podrá avanzar al siguiente módulo.
15. Un estudiante no puede matricular un módulo si registra choque de horarios con otros módulos.
16. Cada módulo para ser abierto debe tener su profesor asignado, y verificar que no haya un choque de horario con otros módulos previamente asignados al profesor.
17. Para todos los programas, es necesario ganar los módulos anteriores, es decir, es necesario ganar los módulos para poder continuar con el programa.
18. También debe contemplar que para abrir un módulo debe tener un laboratorio asignado. Verificar que no haya choques de horario en los laboratorios con otros módulos asignados al laboratorio.
19. A los módulos se les puede asignar los horarios correspondientes y calcular su fecha final, pero se debe asignar un profesor y un laboratorio antes de colocar el módulo como “abierto”, es decir, solo se puede abrir un módulo si tiene profesor y laboratorio previamente asignados.

# Requerimientos funcionales

La base de datos debe ser capaz administrar toda la información de:

1. Estudiantes.
2. Programas.
3. Módulos.
4. Profesores.
5. Certificaciones y títulos de los profesores.
6. Laboratorios/Aulas.

El sistema debe ser capaz de matricular a más de una persona a la vez.

El sistema debe ser capaz de abrir un nuevo módulo con un código independiente, permitiéndole al usuario escoger un horario por día, días de la semana en que se va a impartir, así también como un profesor certificado, un laboratorio, la fecha inicio y fecha final.

El sistema debe ser capaz de llevar la cuenta de cuantos cupos van quedando disponibles dependiendo de la cantidad de estudiantes de hayan matriculado en los módulos disponibles.

El sistema debe ser capaz de permitir que el usuario pueda escoger a su conveniencia los módulos que puede y quiere matricular, así como el horario, el laboratorio y el nombre del profesor que va a impartir el curso.

El sistema debe mostrar el nombre del módulo, el horario, el aula y el nombre del profesor que va a impartir el curso.

El sistema debe únicamente mostrar los módulos disponibles según los programas a los que este matriculado.

El sistema será capaz de calcular la fecha final del módulo, tomando en cuenta los días feriados (no lectivos), los cuales se deben registrar en la Base de Datos al inicio de cada año.

El sistema debe indicar cuando un módulo se quedó sin cupo, no permitiendo matricularlo.

El sistema debe ser capaz de evitar choque de horarios, tanto para los módulos matriculados por estudiantes, como para los profesores, y los laboratorios.

El sistema debe ser capaz de calcular el salario de los profesores, esto mediante un calculo el cual se conforma por el precio de la hora y la cantidad de horas trabajadas, también se toma en cuenta las deducciones del seguro.

La base de datos deberá ser capaz de indicar la hora y fecha de matrícula de cada estudiante dependiendo del programa y del promedio del estudiante.

La base de datos debe indicar si un estudiante no cuenta con los requisitos para los módulos, si no cuenta con los requisitos no va a dejar matricular dichos módulos.

El sistema debe realizar el cobro de la matrícula, según el precio del programa o programas a los que esta matriculando, se realiza mediante la suma de los costos de dichos programas.

El sistema debe ser capaz de registrar los pagos que ya han sido realizados por los estudiantes

El sistema debe ser capaz de verificar las fechas establecidas de la matrícula.

Al formalizar la matrícula, el sistema debe permitir generar un informe descargable con la información de los módulos matriculadas, sus respectivos horarios, créditos, profesores y aulas.

El sistema debe ser capaz de asignar las notas a los estudiantes después de terminar el módulo.

# Reglas de negocio

Por el momento únicamente se van a impartir 3 programas, sin embargo, esto podría cambiar en un futuro.

El precio de la matrícula depende de cuantos módulos matricule el estudiante.

El estudiante debe cancelar en efectivo, transferencia bancaria o con tarjeta de crédito la matrícula en su totalidad antes de iniciar las lecciones, sino no podrá asistir y no va a recibir su nota final si no ha cancelado el módulo, por ende, tampoco podrá avanzar al siguiente módulo.

Los profesores pueden impartir varios cursos siempre y cuando no tengan choque de horario.

Los profesores deben tener una certificación para impartir un determinado módulo. Un profesor puede tener certificaciones para impartir diferentes módulos, mientras que otro sólo podría tener una sola certificación, por lo que sólo podrá impartir un solo módulo en la Universidad.

La oferta académica dependerá de cada programa, ya que no todos los módulos existentes están siempre disponibles.

Cada módulo para ser abierto debe tener su profesor y laboratorio asignado, y verificar que no haya un choque de horario con otros módulos previamente asignados al profesor o al laboratorio.

A los módulos se les puede asignar los horarios correspondientes y calcular su fecha final, pero se debe asignar un profesor y un laboratorio antes de colocar el módulo como “abierto”, es decir, solo se puede abrir un módulo si tiene profesor y laboratorio previamente asignados.

Cada módulo puede tener más de un grupo abierto con su respectivo horario.

Cuando se abre un módulo se solicita al usuario la fecha de inicio. El sistema calculará la fecha final considerando los datos anteriores (días y horarios en que se impartirá), además de los días feriados, los cuales no son lectivos.

Cuando un/a estudiante vaya a realizar la matrícula de un módulo, debe indicar el módulo y el grupo en el que desea matricularse.

Es necesario verificar que no haya choques de horario en los laboratorios con otros módulos asignados al laboratorio.

Cada módulo tiene un requisito o correquisito, un estudiante no puede matricular módulos que tienen requisito de otras, hasta haberlos aprobado.

El horario de la universidad va a ser de 7:00am a 6:00pm de lunes a viernes, por lo tanto, las lecciones deben estar contempladas en ese horario.

Los módulos pueden tener un mínimo de dos horas y un máximo de cinco horas diarias.

Los módulos se pueden impartir: uno, dos, tres, cuatro o cinco días por semana.

Un estudiante puede matricular en más de un programa a la vez, siempre y cuando no choquen en sus módulos y no tengan requisitos pendientes sin aprobar.

Existe un periodo de 3 días para realizar la matrícula de los módulos, pasados los tres días de matrícula el estudiante no podrá matricular hasta que se vuelvan a abrir el proceso de matrícula.

Un estudiante no puede matricular antes de la fecha y hora de la matrícula, sin embargo, después de esta fecha y hora, puede matricular sin ningún problema, siempre y cuando el periodo de matrícula (3 días) no se haya terminado.

El estudiante tendrá 48 horas para realizar el pago de su matrícula, si se sobrepasan las 48 horas su matrícula no se va a formalizar, por lo que no podrá matricular hasta el siguiente semestre

La matrícula se hace en línea, por medio de la página web de la Universidad.

Cada estudiante tiene un usuario y contraseña para realizar su matrícula, dicho usuario se basa en la cédula de cada estudiante.

Los profesores de la universidad ganan un salario, el cual se basa según la cantidad de horas trabajas, el monto por hora y se incluyen las deducciones.

En la matrícula se le cobra a cada estudiante, una póliza del INS.

Cada módulo tiene calificaciones finales, cuya nota mínima es de 70 para continuar en el programa, si no debe repetir el módulo y volver a pagar.

# Modelo Conceptual

Diagrama

Descripción generada automáticamente

# Planteamiento conceptual de la solución.

1. El sistema debe permitir la administración de los ***programas*** que imparte la universidad, así como cada módulo en que están divididos dichos programas.
2. El sistema debe permitir la administración de los ***estudiantes*** de la Universidad, ingresar estudiantes nuevos, eliminar de forma lógica estudiantes existentes que ya aprobaron el programa, modificar los datos de los estudiantes y consultarlos, de esto se encarga un funcionario de la universidad, para ello se requieren los siguientes datos:
   1. Identificación
   2. Nombre
   3. Apellido1
   4. Apellido2
   5. Teléfono 1
   6. Correo electrónico
   7. Distrito
   8. Cantón
   9. Provincia
   10. País
   11. Fecha de Nacimiento
   12. Estado Civil
3. El sistema debe permitir la administración de los ***profesores*** de la Universidad, ingresar profesores nuevos, eliminar de forma lógica los profesores existentes que ya no trabajan, modificar los datos de profesores y consultarlos, para ello se requieren los siguientes datos:
   1. Identificación
   2. Nombre
   3. Apellido1
   4. Apellido2
   5. Teléfono 1
   6. Correo electrónico
   7. Certificaciones
   8. Fecha de Nacimiento
   9. Estado Civil
   10. Distrito
   11. Cantón
   12. Provincia
   13. País
   14. Dirección exacta
4. El sistema debe permitir la administración de los ***horarios*** en los que se imparten los módulos, con sus respectivas horas y días asignados.
5. El sistema debe permitir la administración de todos los ***laboratorios*** en los que se imparten los módulos, con sus respectivas horas y días disponibles.
6. Toda la información generada de los módulos (horario, profesor, aula) se almacena en la base de datos ya que dicha información se requiere para los procesos de la matrícula.
7. Se debe realizar la factura de la matrícula, tomando en cuenta, la estrategia de solución que contempla calcular el precio total de la matrícula de cada estudiante, dependiendo de la cantidad y del precio de módulos que estos matriculen.
8. Al realizar el proceso de la matrícula el sistema deberá capaz de verificar las fechas establecidas.
9. Al realizar el proceso de matrícula el sistema automáticamente deberá comprobar la información de los estudiantes, para que estos no puedan matricular módulos sin tener los requisitos correspondientes.
10. El sistema permite que el usuario asigne a cada módulo, un horario, un profesor y un aula dependiendo del programa de dicho modulo y las disponibilidades de horario, profesores y aulas.
11. Toda la información generada en el proceso de matrícula se almacena en la base de datos, ya que dicha información para la creación de informes con los módulos, horarios, profesores y laboratorios, que los estudiantes pueden descargar para visualizar dicha información.
12. Al realizar el proceso de formalización definitiva de la matrícula el sistema automáticamente deberá comprobar si ya se realizó el pago de esta.

# Modelo lógico

Diagrama

Descripción generada automáticamente

# Diccionario de Datos

|  |  |  |  |
| --- | --- | --- | --- |
| **Matrículas** | | | |
| **Campo** | **Tipo** | **Not Null** | **Observaciones / Restricciones** |
| Num\_recibo | int | X | Llave Primaria, Identity |
| ID\_Estudiante | int | X | Llave foránea |
| Fecha\_Hora | Datetime | X | Predeterminado: Fecha actual |
| Usuario Matricula | Varchar (25) | X |  |
| Monto\_matricula | Decimal(10,2) | X | Restricción: >=0 |
| Estado\_M | Varchar(3) | X | Predeterminado: ACT  Restricción : ACT o INA |

|  |  |  |  |
| --- | --- | --- | --- |
| **Estudiantes** | | | |
| **Campo** | **Tipo** | **Not Null** | **Observaciones / Restricciones** |
| ID\_Estudiante | int | X | Llave Primaria |
| Nombre | Varchar (20) | X |  |
| Apellido1 | Varchar (20) | X |  |
| Apellido2 | Varchar (20) | X |  |
| Teléfono | int | X |  |
| Email | Varchar (20) | X |  |
| Provincia | Varchar (20) | X | Restricciones: San José, Alajuela, Cartago, Heredia, Guanacaste, Puntarenas, Limón. |
| Fecha\_Ingreso | Date |  | >=fecha actual |
| Estado\_Estudiante | Varchar (3) | X | Restricciones: ACT/INACTIVO |
| Borrado | bit | X | Predeterminado = 0 |

|  |  |  |  |
| --- | --- | --- | --- |
| **Detalle Matriculas** | | | |
| **Campo** | **Tipo** | **Not Null** | **Observaciones / Restricciones** |
| Num\_recibo | Varchar (20) | X |  |
| Cod\_modulo\_disponible | Varchar (20) | X | Llave foránea |
| Nota Final | decimal (5,2) |  | >0 y <=100 |
| Estado\_D | Varchar(3) |  | Restricción: 'ACT','RET','APR','REP  Predeterminado: 'ACT' |

|  |  |  |  |
| --- | --- | --- | --- |
| **Módulos\_Disponibles** | | | |
| **Campo** | **Tipo** | **Not Null** | **Observaciones / Restricciones** |
| Cod \_Módulo\_Disponible | int | X | Llave Primaria, identity |
| Cod \_Programa\_ Módulo | Int | X | Llave foránea |
| ID\_Profesor | int | X | Llave foránea |
| Costo\_Módulo | real | X |  |
| Duración\_Modulo | int | X |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Horario** | | | |
| **Campo** | **Tipo** | **Not Null** | **Observaciones / Restricciones** |
| Cod\_horario | int |  | Primary key |
| Cod \_Módulo\_Disponible | int | X | Llave foranea, identity |
| Dia | char (1) | X | días de la semana, siendo 1 el lunes y 5 el viernes |
| Horas\_totales | Time | X |  |
| Hora\_Inicio | Time | X |  |
| Hora\_Final | Time | X |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Profesores** | | | |
| **Campo** | **Tipo** | **Not Null** | **Observaciones / Restricciones** |
| ID\_Profesor | int | X | Llave Primaria, Único |
| Nombre | Varchar (20) | X |  |
| Apellido1 | Varchar (20) | X |  |
| Apellido2 | Varchar (20) | X |  |
| teléfono | int | X |  |
| Email | Varchar (20) | X |  |
| Certificaciones | Varchar (120) |  |  |
| Provincia | Varchar (20) | X | Restricciones: San José, Alajuela, Cartago, Heredia, Guanacaste, Puntarenas, Limón. |
| Borrado | bit | X | Predeterminado = 0 |

|  |  |  |  |
| --- | --- | --- | --- |
| **Módulos** | | | |
| **Campo** | **Tipo** | **Not Null** | **Observaciones / Restricciones** |
| Cod\_ Módulo | int | X | Llave Primaria |
| Nombre \_Módulo | Varchar (20) | X |  |
| Requisitos | Varchar (100) |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Programa\_Módulos** | | | |
| **Campo** | **Tipo** | **Not Null** | **Observaciones / Restricciones** |
| Cod\_Programa\_Módulo | Int | X | Llave Primaria, Identity |
| Cod \_ Módulo | Int | X | Llave foránea |
| Cod \_Programa | Varchar (20) | X | Llave foránea |
| Detalles | Varchar (20DETA0) |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Programas** | | | |
| **Campo** | **Tipo** | **Not Null** | **Observaciones / Restricciones** |
| Cod \_Programa | Varchar (20) | X | Llave Primaria |
| Nombre\_Programa | Varchar (20) | X |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Laboratorios** | | | |
| **Campo** | **Tipo** | **Not Null** | **Observaciones / Restricciones** |
| Cod \_Laboratorio | Int | X | Llave Primaria, identity |
| Cod\_horario | int | X | Llave foránea |
| Cod\_modulos\_disponibles | Varchar(20) | x | Llave foránea |

|  |  |  |  |
| --- | --- | --- | --- |
| **Planillas** | | | |
| **Campo** | **Tipo** | **Not Null** | **Observaciones / Restricciones** |
| Cod \_Planillas | int | X | Llave Primaria,identity |
| ID\_profesor | Varchar (50) | X |  |
| Fecha\_Pago | date | X | Predeterminado: 1> y 5< |
| Horas\_trabajadas | float | X |  |
| Horas\_extra |  |  |  |
| Precio\_hora | float | X |  |
| Salario | real | X |  |
| Deducciones | real | X |  |

# Conclusiones:

En este proyecto del módulo gestión de base de datos, se han puesto en práctica básicamente todos los temas estudiados a lo largo del curso, como lo son: la determinación de las diferentes reglas de negocio y los requerimientos funcionales, también la creación de los modelos conceptuales y lógicos, así como también se debe realizar la identificación de todas las identidades necesarias, sus relaciones y atributos, diccionario de datos siendo todo esto de suma importancia para una buena implementación de la base de datos.

Para la realización de este proyecto se tuvo que investigar y analizar de forma detallada la definición del problema, esto con el fin de tener claro todos los detalles indispensables que deben ser tomados en cuenta para un diseño adecuado de la base de datos.

Es de suma importancia analizar previamente el problema planteado de la clientela, esto con el fin de tener el planteamiento de la solución bien concreto, siendo este bastante claro, completo y consistente, donde se toma en cuenta todo lo fundamental, para a la hora de realizar la implementación de la base de datos hacerlo de la manera más correcta, ordenada, rápida, de forma eficaz y sobre todo manteniendo la integridad de los datos evitando los posibles errores y redundancias.

El diccionario de datos es una herramienta fundamental y de gran ayuda para la creación de la base de datos, se debe siempre tartar de implementar de la mejor manera posible, con datos claros, precisos y correctos.

A la hora de las creaciones de las tablas en la base de datos es fundamental que los datos coincidan con el diccionario de datos, teniendo en cuenta el tipo de datos, las restricciones, los valores predeterminados, y las llaves foráneas y primarias, que siempre deben de coincidir.

Las consultas son fundamentales para poder recapitular todos los registros y sus relaciones en la base de datos, con el fin de hacer más simple y rápido el trabajo, se pueden hacer consultas uniendo todas las tablas, hasta las que no tienen relación entre si .

La integridad referencial es vital para evitar tener un proyecto con datos duplicados, datos faltantes, alterados o incorrectos, por lo que es de suma importancia aplicarla siempre.

Los procedimientos almacenados pueden devolver uno o varios conjuntos de resultados a través de una instrucción “select”.

Un procedimiento almacenado puede devolver valores numéricos enteros a través de la instrucción ”return”, donde se puede comprobar situaciones de si o no existe.