**Instituto Nacional de Aprendizaje**

**Programador de Aplicaciones Informáticas**

**Módulo Gestión de Base de Datos**

**Proyecto Individual**

**Fase I**

**Estudiante: Mariel Daniela Rojas Sanchez**

**Cedula: 208030487**

**Profesor: Luis Alonso Bogantes**

**24 enero 2022**

Índice

[Introducción: 2](#_Toc125399271)

[Objetivo General: 3](#_Toc125399272)

[Objetivos Específicos: 3](#_Toc125399273)

[Definición del problema. 4](#_Toc125399274)

[Conclusiones: 6](#_Toc125399275)

[Bibliografía: 7](#_Toc125399276)

# 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:

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. **Node 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. Materias.
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 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 aula 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 determinar si un estudiante intenta matricular antes de su cita, si esto sucede el sistema no dejara ingresar al sistema a este estudiante hasta que su cita llegue a la fecha y hora indicada.

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

La base de datos deberá ser capaz de indicar la cita 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 de forma automática el cobro de la matricula, según el programa o programas a los que esta matriculando.

El sistema debe ser capaz de registrar los pagos que ya han sido realizados por os 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 determinar si un estudiante es de primer ingreso o estudiante regular, esto para determinar el promedio y calcular la fecha de la cita de matrícula.

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

# Reglas de negocio

Únicamente se van a impartir 3 programas:

1. **Programación de Dispositivos Móviles.**
2. **Programación de Sistemas de Escritorio.**
3. **Programación de Páginas Web.**

El precio de la matrícula depende de cuantos módulos matricule:

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.

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.

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. Node js, duración: 90 horas, costo: ₡185 000.

El sistema debe registrar los pagos realizados por los estudiantes.

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.

Los módulos pueden tener un mínimo de dos horas y un máximo de 5 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.

La matrícula se da con una cita que depende del promedio del estudiante

Un estudiante no puede matricular antes de la fecha y hora de su cita de 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.

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.

Un estudiante no va a recibir la nota final si no ha cancelado el módulo, por ende, tampoco podrá avanzar al siguiente módulo.

.

# Conclusiones:

En este proyecto del módulo de programación estructurada se han puesto en práctica básicamente todos los temas estudiados a lo largo del curso, como lo son: diferentes estructuras de decisión, estructuras cíclicas o de iteración, sentencias de salto, uso de variables constantes, el uso de funciones, de diferentes librerías y bibliotecas, soluciones a diferentes errores de sintaxis, errores de lógica o ejecución, así como también la utilización de diferentes operadores.

Para la realización de este proyecto tuve que investigar bastante diferentes soluciones en cada uno de los pasos de las alternativas propuestas, como por ejemplo para validar algunas opciones de juego; También fue necesario investigar algunas librerías y bibliotecas para implementar en el proyecto.

La ejecución de estos ejercicios que se basan en juegos de mesa es de suma importancia para implementar la lógica, ya que se requiere hacer un pensamiento matemático y lógico para que todo funcione de la mejor manera, con respecto a todas las observaciones, suposiciones y las reglas de negocio.

Las alternativas de solución son de suma importancia para tener una mayor claridad de cómo resolver los ejercicios propuestos y tener dos formas diferentes de resolver un mismo problema, para llegar a la solución más adecuada.

A la hora de implementar los ejercicios, fue esencial la capacidad de analizar y resolver los distintos problemas que se van presentando conforme se van desarrollando, así como el planteamiento tanto de los problemas a resolver, como con el tiempo de organización.

# Bibliografía:

<https://cplusplus.com/>

<https://www.mundijuegos.com.mx/multijugador/domino/reglas/>

<https://www.pokerstars.com/es-419/casino/how-to-play/blackjack/rules/?no_redirect=1>

<https://www.google.com/amp/s/www.mundodeportivo.com/uncomo/ocio/articulo/como-jugar-al-domino-1206.html%3famp=1>

<https://youtu.be/r0bhtmXjDnU>