INA - Centro de Formación Profesional de San Ramón

Programador(a) de Aplicaciones Informáticas

Gestión de Bases de Datos

Proyecto Fase 2

Base de datos para: Instituto Especializado de Entrenamiento en Triatlón (IEET)

Nelson Osvaldo Jiménez Jiménez

Redwin Valverde Castro

Fecha de Entrega: 27/05/2021

# Tabla de contenidos

Contenido

[Tabla de contenidos 2](#_Toc76057149)

[Introducción 3](#_Toc76057150)

[Objetivo principal 4](#_Toc76057151)

[Objetivos Específicos 4](#_Toc76057152)

[Definición del problema 5](#_Toc76057153)

[Análisis y requerimientos funcionales 7](#_Toc76057154)

[Reglas básicas del negocio 8](#_Toc76057155)

[Modelo conceptual de la base de dados 10](#_Toc76057156)

[Planteamiento de nivel conceptual de la solución 11](#_Toc76057157)

[Modelo lógico de la base de datos 13](#_Toc76057158)

[Diccionario de datos 14](#_Toc76057159)

[Conclusión 19](#_Toc76057160)

# Introducción

En el trabajo a realizar a continuación, comprenderá de la primera parte del proyecto de gestión de bases de datos, en el cual se crearán los cimientos de una base de datos, tomando en cuenta el problema dado por el Instituto Especializado de Entrenamiento en Triatlón (IEET), en dicho problema, se intentará llegar a la solución más optima y viable para ambos lados de la moneda, dando solución a todos los requerimientos básicos y otros añadidos con el paso del tiempo en implementación, los más importantes son: la base de datos administra la matrícula de estudiantes con sus respectivos pagos de inscripción y mensualidades, la información de las personas entrenadoras y los programas impartidos. Un programa se imparte a un grupo de atletas, tiene fecha de inicio y una fecha de fin que el sistema debe calcular automáticamente.

Todo esto para llegar a una solución adecuada y optima de dicho problema en el módulo de gestión de bases de datos.

## Objetivo principal

Realizar el diseño de la base de datos de acuerdo al caso proporcionado.

## Objetivos Específicos

1. Realizar el análisis del problema que debe solucionar la base de datos.
2. Determinar los requerimientos funcionales.
3. Determinar las reglas básicas del negocio.
4. Crear el modelo conceptual de la base de datos.
5. Plantear a nivel conceptual la solución.
6. Crear el modelo lógico de la base de datos.
7. Crear el diccionario de datos de la base de datos.

## Definición del problema

Usted debe crear una base de datos para un Instituto que forma atletas en triatlón, se requiere que la base de datos administre la matrícula de estudiantes con sus respectivos pagos de inscripción y mensualidades, la información de las personas entrenadoras y los programas impartidos. Un programa se imparte a un grupo de atletas, tiene una fecha de inicio y una fecha de fin que el sistema debe calcular automáticamente.

La información básica que usted maneja para crear el sistema es la siguiente:

1. Se atiende grupos de un mínimo de 5 personas y un máximo de 15 personas.
2. El programa comprende 4 módulos:
   1. Gimnasio, 80 horas.
   2. Natación, 110 horas.
   3. Maratón, 120 horas.
   4. Ciclismo, 90 horas.
3. Cada entrenamiento diario tiene una duración de dos horas diarias (no fragmentadas), por ejemplo, una persona podría tener su sesión de entrenamiento de 9:00 a.m. a 11:00 a.m., pero no es posible que vaya una hora de 9:00 a.m a 10:00 a.m y en la tarde vaya otra hora de 3:00 p.m. a 4:00 p.m.
4. El sistema debe registrar los pagos mensuales y generar informes para determinar quiénes están al día y quiénes no. El costo de la matrícula es de 15 000 colones, y cada módulo cuesta 125 000 colones.
5. Las personas entrenadoras tienen derecho a 5 semanas de vacaciones al año, las cuales pueden fragmentar y disfrutar en la fecha en que cada persona lo decida. Las vacaciones se ingresan al sistema a inicio de año para cada persona, para poder calcular las fechas. Si una persona cambia sus fechas de vacaciones en el transcurso del año, esto modificará las fechas de los programas que imparte.
6. Las personas entrenadoras pueden tener incapacidades, permisos u otras situaciones no planificadas que también pueden afectar las programaciones, y la base de datos debe ser capaz de recalcular las fechas involucradas.
7. Las personas entrenadores tienen certificaciones en cada una de las 4 áreas, no necesariamente todas las personas entrenadoras tienen certificación para impartir los 4 módulos.
8. Las personas entrenadoras tienen un horario de trabajo definido por días y horas, por ejemplo, una persona entrenadora puede trabajar de lunes a viernes o podría trabajar sólo algunos días a la semana, además tienen un horario, por ejemplo, su horario podría ser de 10:00 a.m. a 2:00 p.m.
9. Cada curso tiene calificaciones finales, cuya nota mínima es de 70 para continuar en el programa.
10. Un estudiante no puede avanzar al módulo siguiente a menos que haya aprobado el módulo anterior.
11. Cada programa nuevo tiene asignado un horario, el cual corresponde a 2 horas diarias. Este horario se mantiene igual para los 4 módulos.
12. La base de datos debe de ser capaz de:
    1. Ingresar la información de los atletas.
    2. Ingresar la información de las personas entrenadoras, y administrar también sus fechas de vacaciones, periodos de incapacidades, licencias u otros.
    3. Crear un programa nuevo a impartir y matricular atletas en un programa nuevo, cada programa tendrá un código independiente.
    4. En un mismo programa, es posible los 4 módulos los imparta la misma persona entregadora o también que sean diferentes personas entrenadoras quienes impartan cada módulo (dependiendo si tienen las certificaciones requeridas, lo cual debe verificar el sistema).
    5. Al abrir un nuevo programa se ingresa: los atletas que formarán el grupo (previamente matriculados y previo pago del curso), la persona instructora que impartirá cada uno de los módulos, el horario en que se ejecutará el programa (debe verificarse que las personas instructoras asignadas no tengan choque de horario con otro curso ya que estén impartiendo, y además que el horario del módulo no esté fuera del horario de la persona docente).
    6. La base de datos calculará automáticamente la fecha de fin de todo el programa para ese grupo.
    7. En caso de presentarse alguna situación especial que afecte las fechas, la base de datos calculará automáticamente el cambio en la fecha final.
13. Estas son las descripciones básicas, cualquier otra información que usted requiera definir para crear el sistema, puede definirla en su análisis y anotarlo como suposiciones en su estrategia.

## Análisis y requerimientos funcionales

1. El sistema será capaz de atender grupos de estudiantes de 5 a 15 personas.
2. El sistema solo contiene 4 módulos, gimnasio, natación, maratón, ciclismo.
3. El sistema deberá administrar las horas de entrenamiento diarias, serán dos horas seguidas por cada sesión de entrenamiento.
4. El sistema debe registrar los pagos mensuales y generar informes para determinar quiénes están al día y quiénes no, matrícula y cada módulo.
5. El sistema podrá agregar nuevas personas entrenadoras si es requerido.
6. El sistema será capaz de calcular las semanas de vacaciones de las personas entrenadoras, se pueden fragmentar y disfrutar en la fecha en que cada persona lo decida, pero solo por semanas. Se ingresan al sistema a inicio de año para cada persona, para poder calcular las fechas. Si una persona cambia sus fechas de vacaciones en el transcurso del año, esto modificará las fechas de los programas que imparte.
7. El sistema puede administrar las incapacidades de las personas entrenadoras, permisos u otras situaciones no planificadas que también pueden afectar las programaciones, y la base de datos debe ser capaz de recalcular las fechas involucradas.
8. El sistema calculará las certificaciones de las personas en cada una de las 4 áreas, no necesariamente todas las personas entrenadoras tienen certificación para impartir los 4 módulos.
9. El sistema calculará el horario de trabajo de las personas entrenadoras, tienen un horario de trabajo definido por días y horas.
10. El sistema **NO** calculará las horas extra.
11. El sistema será capaz de agregar nuevos estudiantes junto con sus datos requeridos.
12. El sistema calculará las calificaciones finales de los atletas.
13. El sistema debe verificar las asistencias de cada estudiante, para verificar si pasa de módulo.
14. El sistema deberá ser capaz de verificar si el estudiante pasó el módulo para avanzar al siguiente.
15. El sistema deberá asignar el horario de cada nuevo programa.
16. La base de datos debe de ser capaz de:
    1. Ingresar la información de los atletas.
    2. Ingresar la información de las personas entrenadoras, y administrar también sus fechas de vacaciones, periodos de incapacidades, licencias u otros.
    3. Crear un programa nuevo a impartir y matricular atletas en un programa nuevo, cada programa tendrá un código independiente.
    4. En un mismo programa, es posible los 4 módulos los imparta la misma persona entregadora o también que sean diferentes personas entrenadoras quienes impartan cada módulo (dependiendo si tienen las certificaciones requeridas, lo cual debe verificar el sistema).
    5. Al abrir un nuevo programa se ingresa: los atletas que formarán el grupo (previamente matriculados y previo pago del curso), la persona instructora que impartirá cada uno de los módulos, el horario en que se ejecutará el programa (debe verificarse que las personas instructoras asignadas no tengan choque de horario con otro curso ya que estén impartiendo, y además que el horario del módulo no esté fuera del horario de la persona docente).
    6. La base de datos calculará automáticamente la fecha de fin de todo el programa para ese grupo.
    7. En caso de presentarse alguna situación especial que afecte las fechas, la base de datos calculará automáticamente el cambio en la fecha final.

## Reglas básicas del negocio

1. Se atiende grupos de un mínimo de 5 personas y un máximo de 15 personas.
2. El programa comprende 4 módulos:
   1. Gimnasio, 80 horas.
   2. Natación, 110 horas.
   3. Maratón, 120 horas.
   4. Ciclismo, 90 horas.
3. Cada entrenamiento diario tiene una duración de dos horas diarias (no fragmentadas), por ejemplo, una persona podría tener su sesión de entrenamiento de 9:00 a.m. a 11:00 a.m., pero no es posible que vaya una hora de 9:00 a.m a 10:00 a.m y en la tarde vaya otra hora de 3:00 p.m. a 4:00 p.m.
4. El sistema debe registrar los pagos mensuales y generar informes para determinar quiénes están al día y quiénes no. El costo de la matrícula es de 15 000 colones, y cada módulo cuesta 125 000 colones.
5. Se puede agregar nuevas personas entrenadoras, incluyendo su número de identificación,
6. Las personas entrenadoras tienen derecho a 5 semanas de vacaciones al año, las cuales pueden fragmentar y disfrutar en la fecha en que cada persona lo decida, pero solo por semanas. Las vacaciones se ingresan al sistema a inicio de año para cada persona, para poder calcular las fechas. Si una persona cambia sus fechas de vacaciones en el transcurso del año, esto modificará las fechas de los programas que imparte.
7. Las personas entrenadoras pueden tener incapacidades, permisos u otras situaciones no planificadas que también pueden afectar las programaciones, y la base de datos debe ser capaz de recalcular las fechas involucradas.
8. Las personas entrenadores tienen certificaciones en cada una de las 4 áreas, no necesariamente todas las personas entrenadoras tienen certificación para impartir los 4 módulos.
9. Las personas entrenadoras tienen un horario de trabajo definido por días y horas, por ejemplo, una persona entrenadora puede trabajar de lunes a viernes o podría trabajar sólo algunos días a la semana, además tienen un horario, por ejemplo, su horario podría ser de 10:00 a.m. a 2:00 p.m.
10. No hay horas extra, ya que cada persona entrenadora tendrá su horario de trabajo definido por día y horas.
11. Cada nuevo estudiante podrá ser ingresado si pagó el módulo y la matrícula, además se ingresará sus datos.
12. Cada curso tiene calificaciones finales, cuya nota mínima es de 70 para continuar en el programa.
13. Si un estudiante falta más de tres veces y no justifica, no podrá avanzar de módulo.
14. Un estudiante no puede avanzar al módulo siguiente a menos que haya aprobado el módulo anterior.
15. Cada programa nuevo tiene asignado un horario, el cual corresponde a 2 horas diarias. Este horario se mantiene igual para los 4 módulos.
16. La base de datos debe de ser capaz de:
    1. Ingresar la información de los atletas.
    2. Ingresar la información de las personas entrenadoras, y administrar también sus fechas de vacaciones, periodos de incapacidades, licencias u otros.
    3. Crear un programa nuevo a impartir y matricular atletas en un programa nuevo, cada programa tendrá un código independiente.
    4. En un mismo programa, es posible los 4 módulos los imparta la misma persona entregadora o también que sean diferentes personas entrenadoras quienes impartan cada módulo (dependiendo si tienen las certificaciones requeridas, lo cual debe verificar el sistema).
    5. Al abrir un nuevo programa se ingresa: los atletas que formarán el grupo (previamente matriculados y previo pago del curso), la persona instructora que impartirá cada uno de los módulos, el horario en que se ejecutará el programa (debe verificarse que las personas instructoras asignadas no tengan choque de horario con otro curso ya que estén impartiendo, y además que el horario del módulo no esté fuera del horario de la persona docente).
    6. La base de datos calculará automáticamente la fecha de fin de todo el programa para ese grupo.
    7. En caso de presentarse alguna situación especial que afecte las fechas, la base de datos calculará automáticamente el cambio en la fecha final.

## Modelo conceptual de la base de dados

## Planteamiento de nivel conceptual de la solución

1. Se creará una tabla para los datos de la institución en caso de que existan más.
2. Cada institución deberá ingresar su información:
3. Número de tienda.
4. Nombre.
5. Teléfono 1.
6. Teléfono 2 (opcional).
7. Correo electrónico 1.
8. Correo electrónico 2 (opcional).
9. Sitio web.
10. Horario.
11. Distrito.
12. Cantón.
13. Provincia.
14. País.
15. Dirección.
16. El sistema debe tomar en cuenta que cada programa tiene diferente cantidad de horas para llevar el control de horarios de cada persona entrenadora:
17. Gimnasio, 80 horas.
18. Natación, 110 horas.
19. Maratón, 120 horas.
20. Ciclismo, 90 horas.
21. Al momento de ingresar la información de una nueva persona entrenadora, el sistema debe solicitar la siguiente información:
    1. Identificación
    2. Nombre
    3. Apellido1
    4. Apellido2
    5. Teléfono 1
    6. Teléfono 2 (opcional)
    7. Cumpleaños
    8. Gimnasio (marcar los que impartirá)
    9. Natación
    10. Maratón
    11. Ciclismo
    12. Correo electrónico 1
    13. Correo electrónico 2 (opcional)
    14. Cuenta bancaria
    15. Distrito
    16. Cantón
    17. Provincia
    18. País
    19. Dirección exacta
22. El sistema debe llevar el control de los horarios de cada persona entrenadora, sus vacaciones, las cuales tiene 5 semanas a favor y deberá elegirlas al inicio del año, y capacitaciones.
23. El sistema no manejará el pago de salarios de los instructores. Solamente registrará las horas de entrada y salida.
24. Al momento de ingresar la información de un nuevo atleta, el sistema debe solicitar la siguiente información:
    1. Identificación
    2. Nombre
    3. Apellido1
    4. Apellido2
    5. Teléfono 1
    6. Teléfono 2 (opcional)
    7. Cumpleaños
    8. Correo electrónico 1
    9. Correo electrónico 2 (opcional)
    10. Distrito
    11. Cantón
    12. Provincia
    13. País
    14. Dirección exacta
25. Un atleta solo puede estar en los entrenamientos si pagó la matrícula y el módulo que llevará.
26. El horario de cada curso lo determina e ingresa la persona administradora.
27. El sistema verifica si el horario de la persona entrenadora no choca con el horario de cada curso.
28. El sistema tomará en cuenta las calificaciones de cada atleta, si su nota es mayor a 70 podrá pasar el módulo, de lo contrario no.
29. El sistema debe generar los grupos dependiendo de que, si el atleta matriculó, pagó el módulo y pasó el módulo anterior.

## Modelo lógico de la base de datos

## Diccionario de datos

|  |  |  |  |
| --- | --- | --- | --- |
| INSTITUTOS | | | |
| **Campo** | **Tipo** | **Not Null** | **Observaciones / Restricciones** |
| ID\_Instituto | tinyint | x | PK, IDENTITY |
| Nombre\_Instituto | varchar(30) | x |  |
| Telefono1\_Instituto | varchar(30) | x |  |
| Telefono2\_Instituto | varchar(30) |  |  |
| Email1\_Instituto | varchar(76) | X |  |
| Email2\_Instituto | varchar(76) |  |  |
| SitioWeb\_Instituto | varchar(150) | X |  |
| Horario\_Instituto | varchar(30) | X |  |
| Distrito\_ Instituto | varchar(30) | X |  |
| Canton\_ Instituto | varchar(30) | X |  |
| Provincia\_ Instituto | varchar(10) | X | Valores: San José, Alajuela, Cartago, Heredia, Guanacaste, Puntarenas, Limón |
| País\_ Instituto | varchar(20) | X |  |
| Direccion\_ Instituto | varchar(150) | x |  |
| Activo\_Instituto | bit | x | Def: 1 |

|  |  |  |  |
| --- | --- | --- | --- |
| ENTRENADORES | | | |
| **Campo** | **Tipo** | **Not Null** | **Observaciones / Restricciones** |
| Cod\_Entrenador | int | x | PK, IDENTITY |
| ID\_Instituto | tinyint | X | FK |
| ID\_Entrenador | varchar(50) | X | UNIQUE |
| Nombre\_ Entrenador | varchar(20) | X |  |
| Apellido1\_ Entrenador | varchar(20) | X |  |
| Apellido2\_ Entrenador | varchar(20) | X |  |
| Nacimiento\_ Entrenador | date | X |  |
| Telefono1\_ Entrenador | varchar(30) | X |  |
| Telefono2\_ Entrenador | varchar(30) |  |  |
| Email1\_ Entrenador | varchar(76) | X |  |
| Email2\_ Entrenador | varchar(76) |  |  |
| Distrito\_ Entrenador | varchar(30) | X |  |
| Canton\_ Entrenador | varchar(30) | X |  |
| Provincia\_ Entrenador | varchar(10) | X | Valores: San José, Alajuela, Cartago, Heredia, Guanacaste, Puntarenas, Limón |
| Pais\_ Entrenador | varchar(20) | X |  |
| Direccion\_ Entrenador | varchar(150) | X |  |
| Cuenta\_Bancaria\_ Entrenador | varchar(30) | X |  |
| Gimnasio\_ Entrenador | bit | X | Def: 0 |
| Natación\_ Entrenador | bit | X | Def: 0 |
| Maraton\_ Entrenador | bit | X | Def: 0 |
| Ciclismo\_ Entrenador | bit | X | Def: 0 |
| Activo\_ Entrenador | bit | x | Def: 1 |

|  |  |  |  |
| --- | --- | --- | --- |
| VACACIONES\_ENTRENADOR | | | |
| **Campo** | **Tipo** | **Not Null** | **Observaciones / Restricciones** |
| Cod\_Entrenador | int | X | FK |
| Inicio\_Vacaciones\_Entrenador | date | X | >= fecha actual |
| Cantidad\_Vacaciones\_Entrenador | tinyint | x | DF: 1 |
| Detalle\_Vacaciones\_Entrenador | varchar(150) | x |  |

|  |  |  |  |
| --- | --- | --- | --- |
| CERTIFICACIONES\_ENTRENADOR | | | |
| **Campo** | **Tipo** | **Not Null** | **Observaciones / Restricciones** |
| ID\_Certificacion | int | X | PK, IDENTITY |
| Cod\_Entrenador | int | X | FK |
| Tipo\_Certificacion | varchar(50) | x |  |
| Descripcion\_Certificacion | varchar(150) | x |  |

|  |  |  |  |
| --- | --- | --- | --- |
| REGISTRO\_HORAS\_LABORADAS\_ENTRENADOR | | | |
| **Campo** | **Tipo** | **Not Null** | **Observaciones / Restricciones** |
| ID\_Registro\_Horas\_Laboradas\_Entrenador | int | x | PK, IDENTITY |
| Cod\_Entrenador | int | X | FK |
| Dia\_Laborado\_Entrenador | date | X |  |
| Detalle\_Dia\_Laborado\_Entrenador | varchar(150) |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| HORARIOS\_ENTRENADOR | | | |
| **Campo** | **Tipo** | **Not Null** | **Observaciones / Restricciones** |
| ID\_Registro\_Horas\_Laboradas\_Entrenador | int | x | FK |
| Entrada1\_Entrenador | time(0) | x |  |
| Salida1\_Entrenador | Time(0) | X |  |
| Detalle\_Horario1\_Entrenador | Varchar(150) | X |  |
| Entrada2\_Entrenador | Time(0) |  |  |
| Salida2\_Entrenador | Time(0) |  |  |
| Detalle\_Horario2\_Entrenador | Varchar(150) |  |  |
| Entrada3\_Entrenador | Time(0) |  |  |
| Salida3\_Entrenador | Time(0) |  |  |
| Detalle\_Horario3\_Entrenador | Varchar(150) |  |  |
| Entrada4\_Entrenador | Time(0) |  |  |
| Salida4\_Entrenador | Time(0) |  |  |
| Detalle\_Horario4\_Entrenador | Varchar(150) |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| INCAPACIDADES\_EVENTOS | | | |
| **Campo** | **Tipo** | **Not Null** | **Observaciones / Restricciones** |
| Cod\_Entrenador | int | X | FK |
| Fecha\_Incapacidad\_Evento | Date | X |  |
| Tipo\_Incapacidad\_Evento | varchar(50) | X |  |
| Dias\_Incapacidad\_Evento | tinyint | X | Def: 1 |
| Detalle\_Incapacidad\_Evento | varchar(150) | X |  |

|  |  |  |  |
| --- | --- | --- | --- |
| ESTUDIANTES | | | |
| **Campo** | **Tipo** | **Not Null** | **Observaciones / Restricciones** |
| Cod\_Estudiante | int | x | PK, IDENTITY |
| ID\_Matricula | int | x | FK |
| ID\_Estudiante | varchar(50) | X | UNIQUE |
| Nombre\_Estudiante | varchar(20) | X |  |
| Apellido1\_Estudiante | varchar(20) | X |  |
| Apellido2\_Estudiante | varchar(20) | X |  |
| Nacimiento\_Estudiante | date | X |  |
| Telefono1\_Estudiante | varchar(30) | X |  |
| Telefono2\_Estudiante | varchar(30) |  |  |
| Email1\_Estudiante | varchar(76) | X |  |
| Email2\_Estudiante | varchar(76) |  |  |
| Distrito\_Estudiante | varchar(30) | X |  |
| Canton\_Estudiante | varchar(30) | X |  |
| Provincia\_Estudiante | varchar(10) | X | Valores: San José, Alajuela, Cartago, Heredia, Guanacaste, Puntarenas, Limón |
| Pais\_Estudiante | varchar(20) | X |  |
| Direccion\_Estudiante | varchar(150) | x |  |
| Observacion\_Estudiante | Varchar(150) |  |  |
| Activo\_Estudiante | bit | x | Def: 1 |

|  |  |  |  |
| --- | --- | --- | --- |
| MATRICULAS | | | |
| **Campo** | **Tipo** | **Not Null** | **Observaciones / Restricciones** |
| ID\_Matricula | int | X | PK, IDENTITY |
| Fecha\_Matricula | Date | X | >= fecha actual |
| Estado\_Matricula | Varchar(8) | X | Valores: en curso, abandono, aprobado, reprobado |
| Nota\_Final | Tinyint |  | Def: 0 |
| Monto\_Cancelado\_Matricula | Smallmoney | X | Def: 0 |
| Tipo\_Cobro\_Matricula | Varchar(20) | X | Valores: curso, curso y matricula |
| Tipo\_Pago\_Matricula | Varchar(15) | x | Valores: transferencia, sinpe, tarjeta, efectivo, otro |
| Detalle\_Matricula | Varchar(150) |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| MODULOS\_ABIERTOS | | | |
| **Campo** | **Tipo** | **Not Null** | **Observaciones / Restricciones** |
| ID\_Modulo\_Abierto | int | x | PK, IDENTITY |
| ID\_Matricula | Int | X | FK |
| ID\_Modulo | Int | X | FK |
| Cod\_Entrenador | int | x | FK |
| Dias\_Modulo\_Abierto | Varchar(50) | X |  |
| Horario\_Modulo\_Abierto | Varchar(50) | X |  |
| Fecha\_Inicio\_Modulo\_Abierto | Date | X | >= fecha actual |
| Detalles\_Modulo\_Abierto | Varchar(150) | X |  |

|  |  |  |  |
| --- | --- | --- | --- |
| MODULOS | | | |
| **Campo** | **Tipo** | **Not Null** | **Observaciones / Restricciones** |
| ID\_Modulo | Int | X | PK, IDENTITY |
| Nombre\_Modulo | Varchar(15) | X |  |
| Duracion\_Modulo | Varchar(10) | X |  |
| Requisitos\_Modulo | Varchar(150) | X |  |
| Detalle\_Modulo | Varchar(150) | X |  |

|  |  |  |  |
| --- | --- | --- | --- |
| MODULOS\_PROGRAMAS | | | |
| **Campo** | **Tipo** | **Not Null** | **Observaciones / Restricciones** |
| ID\_Modulo\_Programa | int | x | PK, IDENTITY |
| ID\_Modulo | Int | X | FK |
| ID\_Programa | int | X | FK |

|  |  |  |  |
| --- | --- | --- | --- |
| PROGRAMAS | | | |
| **Campo** | **Tipo** | **Not Null** | **Observaciones / Restricciones** |
| ID\_Programa | Int | X | PK, IDENTITY |
| Nombre\_Programa | Varchar(15) | X |  |
| Descripcion\_Programa | Varchar(150) | X |  |
| Requisitos\_Programa | Varchar(150) | X |  |
| Observaciones\_Programa | Varchar(150) | X |  |
| Adicional\_Programa | Varchar(150) |  |  |

## Conclusión

Durante la elaboración del trabajo anterior, se llevó a cabo cada una de las clases tenidas con el profesor, mediante el uso de sql server y diferentes aplicaciones para el desarrollo de una base de datos adecuada a todo lo aprendido, usando e investigando cada variable para usarla de buena manera, observando y realizando cada método de normalización.