

**“DISEÑAR UN SISTEMA ESCOLÁSTICO PARA AGILIZAR EL
PROCESO ACADÉMICO Y ADMINISTRATIVO DE LOS
ESTABLECIMIENTOS DE EDUCACIÓN MEDIA.”**

AUTOR:

Vinicio Xavier López Bonilla

**TESIS DE GRADO PRESENTADA COMO REQUISITO PARCIAL PARA OPTAR
AL TÍTULO DE INGENIERO EN INFORMÁTICA Y SISTEMAS
COMPUTACIONALES**

CONVENIO:

UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL “SAN FRANCISCO”
(San Francisco – Argentina)

UNIVERSIDAD TÉCNICA DE COTOPAXI
(Latacunga – Ecuador)



SAN FRANCISCO – CÓRDOBA - ARGENTINA

2008

PÁGINA DE APROBACIÓN DEL TUTOR
“DISEÑAR UN SISTEMA ESCOLÁSTICO PARA AGILIZAR EL PROCESO
ACADÉMICO Y ADMINISTRATIVO DE LOS ESTABLECIMIENTOS DE
EDUCACIÓN MEDIA.”

ESPECIALIDAD:

- INGENIERÍA EN INFORMÁTICA Y SISTEMAS COMPUTACIONALES

ASESOR DEL TRABAJO:

- INGENIERO CÉSAR MINA

CALIFICACIÓN:

NÚMERO

LETRAS

- INGENIERO ROBERTO VIOTTI

CALIFICACIÓN:

NÚMERO

LETRAS

SAN FRANCISCO – CÓRDOBA - ARGENTINA
2008

PÁGINA DE APROBACIÓN DEL TRIBUNAL DE GRADO

“DISEÑAR UN SISTEMA ESCOLÁSTICO PARA AGILIZAR EL PROCESO ACADÉMICO Y ADMINISTRATIVO DE LOS ESTABLECIMIENTOS DE EDUCACIÓN MEDIA.”

APROBADO POR LOS MIEMBROS DEL TRIBUNAL DE GRADO:

FECHA: _____

PÁGINA DE AUTORÍA

El presente trabajo es responsabilidad del autor que lo ha realizado de acuerdo al cronograma de actividades y trabajo presentado, mismo que cumple con los requisitos y exigencias de la investigación científica.

AUTOR

Vinicio Xavier López Bonilla

C.I. 050241001-2

AGRADECIMIENTO

Con el presente documento elevo mi indeleble agradecimiento a todas las personas que hicieron posible la culminación exitosa del mismo, de manera especial a mi tutor Ing. Cesar Mina, que conjuntamente con el Ing. Roberto Viotti me asesoraron durante todo el desarrollo de este proyecto.

De igual manera a la Universidad Técnica de Cotopaxi y a la Universidad Tecnológica Nacional, que por medio del convenio existente entre las mismas, posibilitaron mi participación en el mencionado plan de estudios universitarios.

De la misma forma expreso mi agradecimiento a todos y cada uno de mis familiares, amigos y demás personas que de una u otra manera coadyuvaron a la culminación exitosa del presente trabajo investigativo.

Vinicio López B.

DEDICATORIA

Este trabajo lo dedico, a mis padres Isaél y Charito, a mi tía Lupis quienes fueron el mástil de mi vida, a mis hermanos Mónica y Mauricio impulsores de seguir camino al triunfo, a mi sobrina Isabelita ángel inspirador en los días mas difíciles de mi vida, a la Universidad Técnica de Cotopaxi y a la Universidad Tecnológica Nacional quienes me abrieron las puertas para hacer un profesional con sentido critico al servicio de la sociedad con eficiencia y eficacia cognitiva. A mis abuelitos Sergio Bonilla y Esperanza Miniguano quienes fueron testigos de cada hora, minuto y segundo de estudio para culminar con todo éxito el deseo de ser Ingeniero en Sistemas.

Vinicio López B.

RESUMEN

Con los avances tecnológicos, la necesidad de los países Latinoamericanos se viene marcando en general, con un constante requerimiento de generar alternativas que mejoren los actuales sistemas de administración y manejo de información, en esta línea, la generación de proyectos de este tipo, están convirtiéndose en una posibilidad real de lograrlo.

Considerando que posicionar la tecnología propuesta en este proyecto dentro de un mercado, cada vez mas exigente y más variable, implica generar una serie de cambios dentro las formas tradicionales de administrar la información de los centros educativos de nivel medio, se propone este diseño escolástico, que en su fin procura, analizar las posibilidades de combinar la actividad de los centros educativos de nivel medio con las herramientas de diseño informático en un solo proyecto, pionero en la propuesta, y, que lleva de forma intrínseca, demostrar que es viable su desarrollo como tal, en dichas instituciones.

Así el proyecto contiene cuatro capítulos, el primero, contiene los pasos esquemáticos para desarrollar el estudio propuesto; introducción, formulación y planteamiento del problema, justificación, hipótesis, objetivos, marco teórico, marco conceptual, mismos que determinan los elementos necesarios para llegar a la consecución de los datos que nos permiten diseñar el proyecto, que, conjuntamente con la metodológica propuesta, y el cronograma de actividades, nos permiten medir los avances que se generen, concordando con el diseño mas adecuado para los análisis respectivos, planteados en el segundo capítulo; de la misma manera la etapa investigativa, a la que se hace referencia en el tercer capítulo, permite detallar a profundidad cada uno de los pasos planteados en los dos capítulos anteriores, estos son una herramienta esencial, y permiten ir puliendo las alternativas existentes, considerando tanto a factores externos como internos, pero que de una u otra forma afectan al proyecto en su esencia.

En el capítulo final, se recogen los resultados por medio de un análisis, en el que se deja ver la situación real del proyecto, y su situación potencial, planteando conclusiones y recomendaciones, encaminadas al éxito del mismo.

Se concluye este estudio, con anexos, en donde constan todos los documentos que lo respaldan, y entregan al lector del presente un detalle mas profundo sobre ciertos ítems encontrados en el desarrollo.

SUMMARY

With technological advances, the need for the countries being Latinoamericanos ticking in general, with a constant requirement to generate alternatives to improve existing systems of administration and management of information in this line, the generation of such projects, are becoming a real possibility of doing so.

Where as position the technology proposed in this project within a market, increasingly demanding and more variable, involves creating a series of changes within the traditional ways of managing information of schools mid-level, it is proposed this design Scholastic, that in his view seeks to analyze the possibilities of combining the activity of schools with middle-level design tools running on a single project, a pioneer in the proposal, which takes the form of intrinsic, to show that their development is viable as such, in such institutions.

So the project contains four chapters, the first, contains the outline steps to develop the proposed study; introduction, and formulation of the problem, justification, assumptions, marc, theoretical framework, conceptual framework, which identifies the same elements necessary to reach the attainment of the data that allow us to design the project, which, together with the methodological proposal, and the schedule of activities, allow us to measure the progress being generated, agreeing with the design more suitable for the respective analysis, raised in the second chapter; In the same way the investigative stage, which is referenced in the third chapter, allowing in-depth detail each step raised in the two previous chapters, these are an essential tool, let go and refine existing alternatives, considering both internal and external factors, but one way or another affect the project in its essence.

In the final chapter, the results are collected through an analysis, which shows the real situation of the project and it's potential situation, raising conclusions and recommendations, leading to success.

We concluded this study, with attachments, which contains all the documents that support it, and delivered to the reader of this deeper detail on certain items found in development.

ÍNDICE DE CONTENIDOS

	Pág.
CAPITULO I	
ESQUEMA DE DESARROLLO	
Introducción	11
Formulación y planteamiento del problema	12
Justificación	13
Hipótesis	14
Objetivos	14
Marco teórico	15
Marco conceptual	17
CAPITULO II	
METODOLOGÍA DE LA INVESTIGACIÓN	
Métodos	29
Técnicas	30
Recursos de apoyo	30
Cronograma de actividades	31
CAPITULO III	
DESARROLLO	
Análisis general	32
Análisis del manejo de la información en las Instituciones Educativas de Nivel Medio	32
Análisis de requerimientos del Sistema	33
Desarrollo de la propuesta	34
Fase de diseño	35
Modelo conceptual de datos	37
Modelo físico de datos	45
Script para la creación de la Base de Datos	53
Diseño de la Base de Datos en SQL Server 2005	53

CAPITULO IV

Conclusiones	54
Recomendaciones	54
Bibliografía	55
Anexos	56

ESQUEMA DE DESARROLLO

CAPÍTULO I

I. INTRODUCCIÓN

A través de los años el mundo ha dado grandes cambios en lo que respecta al campo tecnológico y educativo, diariamente apreciamos numerosos ejemplos, así miramos como la manufacturación es remplazada por computadoras, haciendo que todo el trabajo que realizaban varias personas hoy se aplique por medio de una sola, con la ayuda de la informática.

Hay que tomar en cuenta, que la tecnología esta tomando parte muy significativa de la educación y de otras ciencias, modificando y aportando a la calidad de vida de los individuos.

Considerando que, dichos avances científicos y tecnológicos en la actualidad se suceden a pasos acelerados, la educación de nivel medio debe crecer y desarrollarse a la par de esta, buscando para ello herramientas que faciliten, seguridad en el desarrollo de sus procesos tanto inicial como terminal, y así también en el almacenamiento de los datos que estas generan.

Basándonos en que la informática y las ciencias computacionales forman parte de la cotidianidad de estas instituciones, como futuro profesional en este campo me he propuesto diseñar un sistema informático, estructurado de una manera sólida que permita llevar a cabo algunas funciones que se requieren dentro de los procesos de las mismas.

Así, dentro de las instituciones de Educación Media se conoce que una de las principales herramientas que aporta a mantener al día las exigencias de los avances tecnológicos, es la automatización de las actividades administrativas y de curriculum, este razonamiento permite ver la necesidad de **diseñar un sistema escolástico para los establecimientos de Educación Media**, mismo que cumpla con el objetivo de agilizar, facilitar, y modernizar, los procesos para dichos establecimientos promoviendo de esta manera la eficacia y la eficiencia en todas sus actividades.

II. FORMULACIÓN Y PLANTEAMIENTO DEL PROBLEMA

Históricamente, y de forma remarcada en Latinoamérica, se puede asegurar, que los datos que se generan dentro de las instituciones de Educación Media, como las calificaciones de los estudiantes, registros de asistencia, boletines informativos, entre otros, siempre se han manejado de forma manual, y siempre han sido el maestro u otros funcionarios quienes llevan acabo esta labor, dando lugar a que por errores o casos fortuitos esta información este desorganizada, dispersa, o desaparecida.

Por lo que se plantea que lo más viable es proponer la correcta aplicación de una base de datos y un sistema informático, que conjugados con un diseño bien estructurado, permita acceder a la información que se requiera de la forma mas rápida y ordenada posible, teniendo presente que se puede acceder a estos beneficios de una manera sencilla, sin que se ocupe mucho espacio y sin invertir altos costos en equipos.

Con estos preceptos, buscando el bienestar de estas instituciones y la satisfacción de sus usuarios, una de las problemáticas a las que se debe enfrentar, es la carencia de un sistema, que les permita manejar esta información de una forma oportuna, ágil, eficaz y eficiente, por lo que se ha establecido la necesidad de diseñar un sistema escolástico, que se convierta en la herramienta fundamental, que aporte en un muy elevado nivel porcentual a la satisfacción de esta problemática.

El presente estudio, posee características innovadoras y permitirá la elaboración del diseño de un sistema escolástico, que coadyuve a potenciar a las instituciones del nivel medio tal como lo demandan los avances tecnológicos, y las exigencias de la sociedad en la actualidad.

Adicionalmente, con el desarrollo de este tema, se pretende despertar en las nuevas generaciones, el interés por la investigación, sobre todo en las diferentes herramientas informáticas y sus aplicaciones, entiendo de antemano que el futuro viene planteando la necesidad de estar acorde con los avances tecnológicos, puesto que las instituciones necesitan de la ayuda de estas tecnologías, para ofrecer sus servicios de una manera eficaz y eficiente, procurando así obtener un nivel de competitividad optimo, dentro de las exigencias que plantea hoy la sociedad.

III. JUSTIFICACIÓN

El acelerado incremento de los educandos, en las instituciones de Educación Media, es un fenómeno que exige un trabajo mancomunado de las distintas áreas del conocimiento, para poder aplicar soluciones reales y prácticas a las distintas debilidades o desfasajes que se pudieran presentar como fenómenos o problemas propios e inherentes al desarrollo de las mismas.

Así, el aporte de la informática y la computación, va dirigido a ofrecer distintas alternativas para la simplificación laboral y mejora de la eficiencia institucional, por ello se propone la creación de un sistema escolástico que será diseñado con parámetros de fácil aplicación y manipulación, mismo que para las instituciones de Educación Media, será una herramienta que facilitara sus actividades, sin tener que preocuparse por hacer cálculos o tener la información dispuesta en papel, sino mas bien de una forma sencilla y eficaz.

Además, ofrece también un punto de fácil acceso a datos como las calificaciones, informes de conducta, control de asistencia de estudiantes y docentes, entre otros, con descripciones completas de los mismos, como son sus nombres, edades, dirección, nivel académico, estadísticas de rendimiento tanto individuales como colectivo, entre otros, mejorando el nivel de control sobre las actividades que se realizan, lo que elevara el nivel de competitividad tanto interno como interinstitucional.

Considerando también que a través de dicho sistema informático se puedan establecer soluciones rápidas y efectivas a las necesidades de los usuarios tanto padres de familia como personal administrativo y docente de estas instituciones, y como un valor agregado también se logrará mejorar el entorno laboral e imagen publica de las instituciones, tornándose más eficiente, además de convertirse en un excelente punto de enlace con la realidad social que rodea a la institución.

Con estos antecedentes, queda plenamente justificada la necesidad que tienen las instituciones de Educación Media, de poseer un sistema escolástico acorde con los avances tecnológicos, desarrollado con programas y lenguajes computacionales, que permitan un tratamiento seguro y confiable de la información de fácil manejo, y flexible a los requerimientos de sus usuarios.

IV. HIPÓTESIS

¿Es posible diseñar un sistema escolástico para agilizar el proceso académico y administrativo de los establecimientos de Educación Media?

V. OBJETIVOS

OBJETIVO GENERAL

- Diseñar un sistema escolástico para agilizar el proceso académico y administrativo de los establecimientos de Educación Media.

OBJETIVOS ESPECÍFICOS

- Obtener información primaria y secundaria de los establecimientos de Educación Media para establecer un diagnóstico situacional
- Análisis y clasificación de la información y datos obtenidos.
- Aplicación de procedimientos y mecanismos para la elaboración del diseño relacional.

VI. MARCO TEÓRICO

El desarrollo mundial, es un proceso inexorable y si bien hoy está focalizado en los movimientos de capitales, información y tecnologías, se está expandiendo a la movilidad de recursos humanos, especialmente, en el marco de la transferencia internacional de conocimientos y tecnologías.

Las tecnologías de la Información y Comunicación deben ser vistas como herramientas de conocimiento, más que herramientas de información.

Estas tecnologías generan significativos cambios, incorporados a las organizaciones de gestión tanto pública como privada.

El impacto se produce en distintos grados. A medida que crece el nivel de cambio que se pretende introducir, aumentan las posibilidades de hallar una solución novedosa y que genere una ventaja competitiva para la organización.

Sin embargo también crecen las dificultades al aplicar estas transformaciones y los riesgos de un posible fracaso.

En general se advierte que las mayores dificultades provienen de los factores organizacionales antes que los estrictamente técnicos.

Esto obliga a presentar una nueva perspectiva para las actividades de los especialistas en sistemas de información, que deben desempeñar un papel activo en la definición de la estrategia de la organización, y en el rol de la tecnología para alcanzar ventajas competitivas.

En relación con lo anterior, se requiere definir una estructura de la organización acorde y poder incorporar exitosamente soluciones informáticas en el ámbito de la organización.

Las instituciones educativas, no constituyen excepciones a estos procesos.

Sucesivas reformas educativas propulsaron la autonomía de los Institutos de Formación, que constituyen organizaciones de Educación Media.

Ese modelo autónomo propició que cada establecimiento adoptara estructuras funcionales distintas y que gestionara sus recursos.

Esta descentralización es entendida como elemento favorecedor de la adaptación de las instituciones al entorno en que se insertan y de la optimización de sus recursos.

En el actual contexto definido por la nueva economía, la innovación, es el resultado de la articulación entre la creación del conocimiento y la institución, y la difusión de este conocimiento en las demás ramas de actividad.

Con esta perspectiva, la idea debe posicionarse en el centro de la innovación, ya que la creación y difusión de conocimiento serán mucho más eficientes, si se garantiza su carácter universal, del cual la misma debe seguir siendo el máximo exponente, mediante la implementación de acciones de transferencia y desarrollo.

Atendiendo a lo expresado en los párrafos precedentes, se presenta en este trabajo una etapa del diseño de un sistema de información tendiente a automatizar la gestión de la información de las instituciones de educación media.

En síntesis esta idea macro tiene bases solidas muy bien planteadas y cimentadas en la necesidad del avance tecnológico que se requiere en los establecimientos de educación media de la ciudad de San Francisco, específicamente hablando, abre las perspectivas de forma muy amplia sobre ítems que antes no se podía controlar ni medir, y que ahora indirectamente afectaran sobre el desarrollo disciplinario y académico de los estudiantes y docentes, conjuntamente con otras responsabilidades de los demás involucrados en el proceso educativo.

Convirtiéndose de esta manera en una herramienta que marcara la pauta para el desarrollo futuro de las aplicaciones de esta área del conocimiento, en pos del desarrollo de las actividades de la sociedad en general, pronosticando de esta manera mejores días para los conglomerados humanos, en donde se genere estos emprendimientos tecnológicos.

VII. MARCO CONCEPTUAL.-

SISTEMA DE INFORMACIÓN

Conjunto u ordenación de elementos necesarios para llevar a cabo algún método, procedimiento o control mediante el proceso de información.

ANÁLISIS Y SISTEMAS

El análisis y diseño de sistemas se refiere al proceso de examinar la situación de una empresa con el propósito de mejorar con métodos y procedimientos más adecuados. El desarrollo de sistemas tiene dos componentes.

Análisis Es el proceso de clasificación e interpretación de hechos, diagnóstico de problemas y empleo de la información para recomendar mejoras al sistemas; especifica que es lo que el sistema debe hacer.

Diseño: Especifica las características del producto terminado; establece como alcanzar el objetivo.

ELEMENTOS DE UN SISTEMA DE INFORMACIÓN

SOFTWARE. Los programas de computadoras, las estructuras de datos y la documentación asociada, que sirve para realizar la metodología.

HARDWARE: Los dispositivos electrónicos que proporcionan la capacidad de computo y las funciones para el mundo exterior.

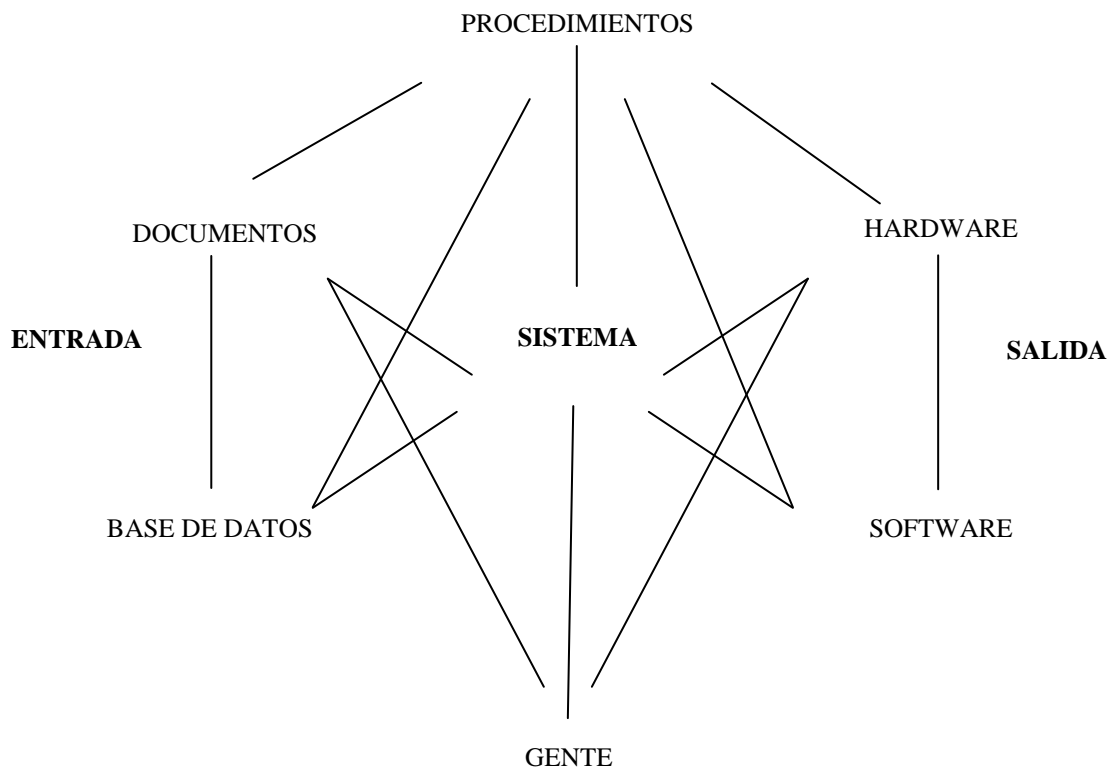
GENTE: Los individuos que son usuarios y operadores del software y del hardware.

BASES DE DATOS: Una colección grande y organizada de información a la que se accede mediante el software y que es una parte integral del funcionamiento del sistema.

DOCUMENTACIÓN: Los manuales, los impresos y otra información descriptiva que explica el uso y / o la operación.

PROCESAMIENTOS: Los pasos que definen el uso específico de cada elemento del sistema o el contexto procedimental en que reside el sistema.

CONTROL: Los sistemas trabajan mejor cuando operan con niveles de control tolerables de rendimiento; ejemplo: la seguridad al acceso de la información de la base de datos.



CLASIFICACIÓN DE LOS SISTEMAS DE INFORMACIÓN

ABIERTOS. Son los que intercambian información, materiales y energía con su ambiente.

CERRADOS. Son de auto contenido, no interactúan con el medio ambiente.

PROBABILÍSTICO. No se conoce con certeza su comportamiento.

DETERMINATIVO. Cualquier estado futuro que adopten puede preverse con antelación.

CARACTERÍSTICAS DE SISTEMA DE INFORMACIÓN

Sus principales características son:

- Suelen lograrse ahorros significativos de mano de obra.
- Son el primer tipo de sistemas de información que se implanta en las organizaciones.
- Son intensivos en entradas y salidas de información; sus cálculos y procesos suelen ser simples y poco sofisticados, requieren mucho manejo de datos para poder realizar sus operaciones y como resultado generan también grandes volúmenes de información.
- Tiene la propiedad de ser recolectores de información.

- Son adaptables de aplicación que se encuentran en el mercado.

Ejemplos: facturación, nóminas, cuentas por cobrar, cuentas por pagar, escolásticos, entre otros.

INTERFACES GRÁFICAS, REPORTE Y CONSULTAS

Facilidad para explorar la información a través de graficas de alta calidad y reportes que se diseñan y obtienen en intervalos cortos de tiempo, la disponibilidad de lenguajes de muy alto nivel para facilitar la consulta de información que contienen las bases de datos.

BASE DE DATOS CORPORATIVA

Integra toda la información de la institución, la cual pueden consultar los diferentes usuarios para construir y utilizar herramientas para lo que requieran.

FUNCIONES DE UN SISTEMA DE INFORMACIÓN

- Reservaciones aéreas.
- Departamento de registro hospitalario.
- Control escolar.
- Preparación de nóminas en operaciones bancarias.
- Sistema de intercomunicación electrónica.

Entre otras.

Para cumplir sus propósitos los sistemas interactúan con su medio ambiente es decir cualquier entidad que se sitúa fuera de los límites del sistema (sistemas abiertos).

Los niveles de rendimiento se miden con estándares. Los rendimientos reales se comparan contra los estándares que vienen de la experiencia.

Las actividades que están por encima o debajo de los estándares deben anotarse, estudiarse y ajustar.

Retroalimentación. La información suministrada a través de la comparación de los resultados con los estándares y el informe de los elementos de control sobre las diferencias. Ejemplo: Una empresa que produce pasteles caros y de baja calidad, las ventas bajas serían la retroalimentación.

Hay que tomar en cuenta los siguientes ejemplos de datos relevantes relacionados con el sistema.

1. **Canales no formales.** ¿Qué interacciones existen entre el personal y los departamentos, pero que no aparecen en el organigrama o en el procedimiento de operación previamente establecido?
2. **Interdependencia.** ¿En qué otras áreas, departamentos y componentes de la empresa se encuentra una dependencia específica? Esto es, ¿Quiénes dependen de quien para realizar sus funciones?
3. **El personal clave.** ¿Cuáles individuos y elementos del sistema son más importantes para su existencia? ¿Cree que un área a la que no se le haya tomado en cuenta su opinión, hubiera aportado buenas ideas al diseño de elaboración del sistema sobre todo si tiene directamente que ver con su función?
4. **Relaciones críticas de comunicación.** ¿Cómo circula la información y las instrucciones entre los componentes de la empresa y como interactúan las diferentes áreas?

HERRAMIENTAS DE DISEÑO

Power Designer

Es una herramienta de modelado colaborativo empresarial producido por Sybase. Power Designer corre bajo Microsoft Windows en modo nativo, y también corre bajo Eclipse como un plugin.

Funcionalidades

Power Designer incluye las siguientes funcionalidades:

- Análisis de Requerimientos
- Modelado de Objetos (UML 2.0 diagrams)
- Modelado de Datos (compatible con la mayor parte de Bases de datos RDBMS)
- Modelado de Procesos de Negocios (ProcessAnalyst) soporta BPMN
- Soporta modelado XML XML Schema y DTD estándares

- Modelado de Data Warehouse (Warehouse Architect)
- Generación de código (Java, C#, VB .NET, Hibernate, EJB3, NHibernate, JSF, WinForm (.NET y .NET CF), PowerBuilder, ...)
- Generación de Reportes
- Repositorio
- Visual Studio 2005 addin
- Eclipse plugin

Principales Características:

- Análisis de Negocio: Modelamiento avanzado de procesos de negocio con vínculos a simulación.
- Ejecución de Procesos
- Modelamiento de Datos: Modelamiento de datos 7 multi nivel conceptual, lógico y de bodega de datos.
- Soporte Abierto a Bases de Datos: Soporte al ciclo de vida para más de 45 RDBMS, incluyendo Oracle 9i, IBM DB/2, MS SQL Server 2000, Sybase ASE, ASA e IQ, MySQL, y muchas más...
- Modelamiento de Objetos: Soporte a modelamiento UML altamente extensible, soportando los 9 diagramas UML.
- Soporte Abierto a Lenguajes de Objeto: Soporte al ciclo de vida para Java, C#, VB-Net, PowerBuilder, XML, C++, y más...
- Potente Productividad: Soporte mejorado a J2EE, .Net y Servicios Web.
- Modelamiento Empresarial: Poderosa tecnología de encadenamiento y sincronización para un poderoso manejo de meta-datos. Técnicas de asociación Objeto/Relacional, XML, de base de datos y de bodega de datos.
- Arquitectura Orientada a Servicios: Orquestación de servicios a través del encadenamiento de componentes UML, UDDI y Servicios Web descritos con WSDL, alrededor de un proceso ejecutable.
- Generación de Documentación: Poderoso generador de reportes HTML y RTF.
- Fácil de Usar: Asistentes mejorados para mayor productividad.

Power Designer ofrece innovaciones en modelamiento de procesos empresariales, incluyendo soporte de simulación y procesamiento ejecutable de procesos de negocio, mayor integración con los ciclos de desarrollo de lenguaje como C# y VB.Net y técnicas más poderosas de generación Inter.-modelos, encadenamiento y sincronización.

Dentro de las nuevas características que brinda esta herramienta, se encuentran:

- **Modelamiento de datos:** Power Designer soporta modelos Conceptuales, Lógicos y Físicos, incluyendo extensiones de modelamiento para bodegas de datos.
- **Modelamiento de aplicaciones:** Power Designer soporte diagramas UML y ofrece asociación avanzada objeto/relacional para manejo e implementación de persistencia. Power Designer también soporta técnicas de modelamiento específicas para XML, encadenadas a modelos UML y de datos.
- **Modelamiento de Procesos de Negocio:** Power Builder soporta diagramas de descripción y definición intuitiva y no técnica de procesos de negocio, así como lenguaje de modelamiento de ejecución de procesos.

TÉCNICAS DE PROGRAMACIÓN

Bases de datos relacionales

Para la realización de las diferentes tablas a utilizarse en el sistema se lo realizo en SQL SERVER 2005.

SQL SERVER 2005

SQL Server 2005 es una plataforma global de base de datos que ofrece administración de datos empresariales con herramientas integradas de inteligencia empresarial (BI). El motor de la base de datos SQL Server 2005 ofrece almacenamiento más seguro y confiable tanto para datos relacionales como estructurados, lo que le permite crear y administrar aplicaciones de datos altamente disponibles y con mayor rendimiento para utilizar en su negocio.

El motor de datos SQL Server 2005 constituye el núcleo de esta solución de administración de datos empresariales. Así mismo, SQL Server 2005 combina lo mejor en análisis, información, integración y notificación. Esto permite que su negocio cree y despliegue soluciones de BI rentables que ayuden a su equipo a incorporar datos en cada rincón del negocio a través de tableros de comando, escritorios digitales, servicios Web y dispositivos móviles.

La integración directa con Microsoft Visual Studio, el Microsoft Office System y un conjunto de nuevas herramientas de desarrollo, incluido el Business Intelligence Development Studio, distingue al SQL Server 2005. Ya sea que usted se desempeñe como encargado de desarrollo, administrador de base de datos, trabajador de la industria de la información o dirija una empresa, SQL Server 2005 ofrece soluciones innovadoras que le ayudan a obtener más valor de sus datos.

Microsoft ha escuchado sus comentarios, investigado en la industria, desafiado a los equipos de Microsoft Research en todo el mundo y pensado creativamente para producir SQL Server 2005, que introduce cientos de nuevas y mejores características. Estas características le ayudarán a progresar en su negocio en tres áreas clave:

Hoy en día las organizaciones enfrentan numerosos desafíos de datos, tales como la necesidad de tomar decisiones más rápidas y más orientadas a datos, la necesidad de aumentar la productividad y flexibilidad del personal de desarrollo y presionan para reducir los presupuestos generales de informática (IT) a la vez que escalan la infraestructura para satisfacer las exigencias cada vez mayores. SQL Server 2005 está diseñado para ayudar a las empresas a enfrentar estos desafíos. Esta solución de administración y análisis de datos de próxima generación ofrece seguridad, escalabilidad y disponibilidad mayores a las aplicaciones de datos empresariales y analíticas, a la vez que las hace más fáciles de crear, desplegar y administrar.

Con la ampliación de las ventajas de SQL Server 2000, SQL Server 2005 ofrece una solución integrada de administración y análisis de datos que ayuda a las organizaciones de cualquier magnitud a realizar lo siguiente:

- Crear, desplegar y administrar aplicaciones empresariales más seguras, escalables y confiables.
- Maximizar la productividad de IT mediante la reducción de la complejidad y el soporte de aplicaciones de bases de datos.
- Compartir datos en múltiples plataformas, aplicaciones y dispositivos para facilitar la conexión de sistemas internos y externos.
- Controlar los costes sin sacrificar el rendimiento, la disponibilidad, la escalabilidad o la seguridad.

SQL Server 2005 potencia su infraestructura de datos en tres áreas clave: administración de datos empresariales, productividad del encargado del desarrollo e inteligencia empresarial (BI). También abre nuevos caminos en precios y licencias accesibles, rutas de actualización a SQL Server 2005 y el sistema Microsoft Windows Server.

Características principales de SLQ Server 2005

Característica Descripción

- **Mirroring de Bases de Datos.** Microsoft SQL Server 2005 amplía las posibilidades de duplicación de logs (“log shipping”) proporcionando a los administradores de BBDD la opción de mirroring. Los administradores pueden usar esta funcionalidad para garantizar la disponibilidad de sus sistemas SQL mediante la configuración de un servidor en espera para su activación automática en caso de fallo (failover).
- **Operaciones de Indexación Online.** La opción de indexado online permite modificaciones concurrentes (actualizaciones, borrados e inserciones) en las tablas subyacentes o datos con índices cluster y de cualquier índice asociado durante la ejecución de DDL de indexación. Por ejemplo, mientras se está reconstruyendo un índice cluster, se puede seguir haciendo actualizaciones a los datos y consultas sobre estos datos.
- **Nuevas herramientas integradas.** SQL Server 2005 introduce el SQL Server Studio, una suite de herramientas de gestión integradas. Este nuevo conjunto incluye nuevas funcionalidades para desarrollo, implantación y resolución de

problemas de Bases de Datos SQL, así como mejoras de las funcionalidades anteriores.

- **Aislamiento de Imágenes (SI).** Se dispone de un nuevo nivel de aislamiento de imagen de BBDD (“Snapshot Isolation”, SI) a nivel de base de datos. SI permite a los usuarios acceder a la última operación realizada utilizando una vista transitoria consistente de la Base de Datos. Esta funcionalidad ofrece una escalabilidad mayor para implementaciones de bases de datos extremadamente grandes.
- **Particionado de Datos.** El particionado de datos se ha mejorado con particiones nativas de tablas e índices, posibilitando soluciones de escalabilidad horizontal. Al particionar tablas de bases de datos muy grandes, el rendimiento de las consultas a la base de datos se mejora notablemente.
- **Backups duplicados (“mirrored backup”).** SQL Server 2005 incluye un nuevo soporte para volúmenes de backup espejados, aumentando la disponibilidad de las copias de seguridad de SQL Server. La posibilidad de replicar el backup permite resolver posibles problemas de corrupción del medio físico de copia.
- **Restauración online .** SQL Server 2005 permitirá realizar una operación de recuperación mientras una instancia de SQL Server está activa. La restauración online mejora la disponibilidad de SQL Server, ya que únicamente los datos que se están recuperando quedan como no disponibles. El resto de la base de datos permanece online y disponible.
- **Recuperación rápida.** Esta característica mejora la disponibilidad de las bases de datos de SQL Server con una nueva opción. Los usuarios podrán reconectarse a una base de datos en recuperación después de que realizar un “roll forward” sobre el log de transacciones.
- **Conexión de Administrador dedicada.** SQL Server 2005 introduce la conexión de administración dedicada, que pueden utilizar los administradores de BBDD para acceder a un servidor en explotación aun cuando el servidor está bloqueado o no disponible por cualquier motivo. Así, los administradores podrán ejecutar funciones de diagnóstico, o sentencias Transact SQL, a fin de poder resolver problemas en el servidor.
- **Mejoras en la Replication.** Para bases de datos distribuidas móviles, SQL Server 2005 proporciona una serie de funcionalidades de replicación extremo a extremo,

incluyendo la posibilidad de publicar bases de datos Oracle. SQL Server 2005 incluirá nuevas mejoras a las herramientas y sobre la escalabilidad de la replicación también.

Características de Desarrollo

- **Soporte para .NET Framework.** SQL Server 2005 introduce la posibilidad de desarrollar objetos de base de datos en lenguajes .NET. Pueden crearse Objetos de código, incluyendo Funciones, Procedimientos y Triggers en lenguajes como C# y VB.NET. Se pueden crear también dos nuevos objetos en estos lenguajes: User-Defined Types y Agregados.
- **Tecnologías XML.** Las tecnologías XML son un estándar importante y ampliamente aceptado para distribuir datos entre diferentes aplicaciones sobre redes locales e Internet. Microsoft SQL Server 2005 soporta el almacenamiento nativo de documentos XML, especificación de consultas con XML y devolución de resultado de consultas como documentos XML.
- **ADO.NET Versión 2.0.** Desde el nuevo soporte para tipos SQL a “Múltiple Active Result Set”, ADO .NET supone una evolución de los métodos de acceso y manipulación de datos, para conseguir una mayor escalabilidad y flexibilidad.
- **Mejoras en Transact-SQL.** SQL Server 2005 introduce muchas posibilidades nuevas para el desarrollo de aplicaciones de bases de datos escalables. Estas mejoras incluyen el manejo de errores, nuevas posibilidades de consultas recursivas y soporte para nuevas funcionalidades del motor de SQL Server.
- **Gestor de Servicio SQL.** El Gestor de Servicio SQL (SQL Service Bróker) ofrece un marco para aplicaciones distribuidas orientados a aplicaciones de línea de negocios a gran escala.
- **Servicios de Notificación.** Los Servicios de Notificación (“Notification Services”) permiten construir aplicaciones de notificación mejoradas, capaces de expedir información personalizada en el momento en que se genera, como puede ser alertas de cotizaciones de bolsa, nuevas suscripciones, alertas de envío de paquetes, o cambios en las tarifas de billetes de avión, a cualquier dispositivo y a millones de suscriptores.

- **Servicios Web.** En SQL Server 2005, los desarrolladores pueden crear servicios Web en la capa de base de datos, convirtiendo a SQL Server en un servidor HTTP. Esta capacidad supone un nuevo modelo de acceso a los datos para aplicaciones basadas en Web Services.
- **Soporte para Xquery.** SQL Server 2005 incluye funcionalidades Xquery de altas prestaciones, que permiten la manipulación de objetos XML en la capa de datos, así como un nuevo juego de herramientas de creación de Xquery.
- **Mejoras en la Búsqueda de Texto Completo.** SQL Server 2005 incluirá soporte para aplicaciones de texto completo ampliadas. Las funcionalidades de catálogo se han mejorado para proporcionar una mayor flexibilidad sobre el conjunto de datos que se catalogan. El rendimiento de las funciones de consulta y la escalabilidad han mejorado sensiblemente.

Una serie de nuevas herramientas de gestión permiten un mayor control de la implementación de texto completo.

- **Mejoras en Seguridad.** SQL Server 2005 incorpora un nuevo modelo de seguridad que separa a los usuarios de los objetos, proporciona un acceso muy granular y un mejor control de los accesos a los datos. Además, todas las tablas del sistema se implementan ahora como Vistas, lo que redundará en un mayor control sobre los objetos de sistema de la Base de Datos.

Objetivos del Diseño de SQL Server

Los clientes están buscando soluciones para sus problemas de negocios. La mayoría de las "soluciones" de bases de datos solamente traen múltiples niveles de costos y complejidad. La estrategia de Microsoft es la de hacer que SQL Server sea la base de datos más fácil de utilizar para construir, administrar e implementar aplicaciones de negocios. Esto significa tener que poner a disposición un modelo de programación rápido y sencillo para desarrolladores, eliminando la administración de base de datos para operaciones estándar, y suministrando herramientas sofisticadas para operaciones más complejas.

Los clientes invierten en sistemas de administración de bases de datos, en forma de aplicaciones escritas para esa base de datos y la educación que implica para la

implementación y administración. Esa inversión debe protegerse: a medida que el negocio crece, la base de datos deberá crecer y manejar más datos, transacciones y usuarios. Los clientes también desean proteger las inversiones a medida que escalan aplicaciones de base de datos hacia equipos portátiles y sucursales.

Para cumplir con estas necesidades, Microsoft ofrece un motor de base datos único que escala desde una computadora portátil que ejecuta Windows® 95 o Windows 98, hasta clúster de procesadores múltiples simétricos de terabyte que ejecutan Windows NT Server Enterprise Edition. Todos estos sistemas mantienen la seguridad y confiabilidad que exigen los sistemas de negocios de misión crítica.

Mientras los sistemas de procesamiento siguen siendo un componente clave para las infraestructuras de bases de datos corporativas, las compañías también están invirtiendo bastante en mejorar la comprensión que tienen de sus datos. La estrategia de Microsoft consiste en reducir el costo y la complejidad del data warehousing mientras hace que la tecnología sea más accesible a una mayor cantidad de público.

CAPITULO II

METODOLOGÍA DE LA INVESTIGACIÓN

VIII. MÉTODOS

La metodología para este estudio, parte de un esquema, acoplado a los requerimientos y necesidades del mismo. El esquema contiene los siguientes puntos principales:

EL MÉTODO CIENTÍFICO.

Entendido como el conjunto de medios empleados en forma lógica para lograr un determinado fin, es la forma de reproducir el conocimiento de un objeto que se estudia o se investiga con fundamentos científicos, se vale de:

- **OBSERVACIÓN.-** Realiza la observación cuantitativa y cualitativa del objeto que se estudia o investiga cuyos elementos están íntimamente relacionados.
- **ANÁLISIS.-** En realizar un estudio detenido de sus elementos que lo componen poniendo en énfasis sus distintas facetas en forma analítica crítica.
- **SÍNTESIS.-** Teorizar la reconstrucción de los hechos o la estructura del objeto sobre la base de la relación del análisis.

MÉTODO HIPOTÉTICO DEDUCTIVO.

Parte de la hipótesis. Y forma parte del tratamiento de las necesidades de investigar y descubrir el por qué de los acontecimientos.

A demás se caracteriza por el estudio y el análisis del contenido teórico-práctico en la relación sujeto objeto.

MÉTODO ANALÍTICO -SINTÉTICO.

Los datos recolectados se los estructura en un análisis de los acontecimientos para expresarlos en una síntesis detallada de invención científica.

MÉTODO DESCRIPTIVO.

Describe las principales causa del objeto a investigar, determina el por qué de los acontecimientos y lo presenta en una composición de ideas secuenciales llamado informe final, justificando el trabajo realizado.

IX. TÉCNICAS

- La observación
- Investigación bibliográfica
- Internet
- Intercambio de ideas

X. RECURSOS DE APOYO

- Asesores
- Instituciones de educación media.

XI. CRONOGRAMA DE ACTIVIDADES

CRONOGRAMA DE TRABAJO	CRONOGRAMA MENSUAL					
	MES 1	MES 2	MES 3	MES 4	MES 5	MES 6
ACTIVIDAD						
1. Elaboración del anteproyecto	XX					
2. Presentación del anteproyecto de tesis	X					
3. Estructuración del equipo de trabajo.		XX				
4. Análisis de los requerimientos de la institución			XX			
5. Análisis del manejo de la información			XX			
6. Desarrollo del Diseño Modelado de Base de Datos				XXXXXXXXXX		
7. Creación del Script de Base de Datos					X	
8. Defensa del proyecto de tesis						

CAPITULO III

XII. DESARROLLO

Análisis General

Realizado el análisis respectivo para el diseño de un sistema escolástico para las instituciones de educación media, con el fin de ayudar en la Eficiencia y Eficacia de los procesos académicos administrativos, se examino la forma como se desarrollaban los procesos mencionados manualmente, mediante el análisis de los documentos que manejan las instituciones educativas de nivel medio.

Las instituciones educativas, deben ser complementadas con los avances tecnológicos existentes para que de esta manera se pueda brindar un mejor servicio a los usuarios de la misma, y de cubrir las falencias que se dan actualmente como es la acumulación de tiempo en los procesos.

Análisis del manejo de información en las Instituciones Educativas de Nivel Medio

Una vez realizado un estudio de las necesidades y del problema que tienen las Instituciones Educativas al momento de ejecutar los procesos, se ha relacionado con la información que se maneja en el departamento de Secretaria, y podemos identificar que el servicio que brinda no cumple a cabalidad con las exigencias y requerimientos de los clientes, debido a que la información se lo maneja en hojas, o archivadores, ocasionando inconvenientes en el momento de buscar y entregar información, es por ello que se ha visto en la necesidad de la creación de una herramienta que permita desarrollar las actividades de la institución de una manera fácil y eficiente, mejorando la administración de la información de toda la Institución.

Análisis de requerimientos del sistema.

Durante el transcurso del desarrollo de este proyecto se fueron ajustando y adaptando los mecanismos de acuerdo a las variables que se presentaban, demostrando que esta metodología brinda una gran flexibilidad y versatilidad para adaptarse a la coyuntura que se presente.

El diseño desarrollado no sólo cumple con el objetivo meramente tecnológico y modernista de comenzar a dotar a los procesos educativos clásicos de las tecnologías actualmente disponibles en el mercado, sino que también se tradujeron en:

- Mejora en las relaciones interpersonales entre la Institución y los Usuarios
- Implicación activa de los participantes
- Mayor facilidad para los procesos que se realizaban manualmente en la Institución

Diversas dependencias de las instituciones educativas afrontaron desde perspectivas diferentes el problema de la organización del trabajo y la sistematización de los procesos administrativos, tratando de incorporar hasta donde los recursos y el contexto lo permitían, ingredientes de automatización y control de los flujos de información. El crecimiento de la institución educativa y la importancia de utilizar eficientemente sus recursos crean la necesidad de sistematizar y centralizar los servicios administrativos.

Nuestro objetivo es entender los detalles del análisis y el diseño para el control de la información que manejan diariamente las Instituciones Educativas de Nivel Medio, las que se detallan a continuación, y las cuales cubren el diseño desplegado.

PROCESOS GENERALES

REFERENCIA	FUNCIÓN
Ref.1	Registros para el seguimiento y control asistencia.
Ref.2	Acceso a notas, materias, horarios y planes de estudio.
Ref.3	Almacenar la información de los estudiantes y personal de la institución.
Ref.4	Generación de partes.

PROCESOS ESPECÍFICOS

ALUMNOS

REFERENCIA	FUNCIÓN
Ref.1	Alta - Modificación del Alumnado.
Ref.2	Almacenamiento de datos personales del alumnado.
Ref.3	Alta - Modificación de notas.
Ref.4	Emisión de certificados y partes.
Ref.5	Control de asistencia, aplicación de sanciones.
Ref.6	Asignación de materias a recibir.
Ref.7	Asignación de horarios.

PERSONAL

REFERENCIA	FUNCIÓN
Ref.1	Alta – Modificación del personal.
Ref.2	Asignación de cargos al personal.
Ref.3	Control de asistencia, aplicación de sanciones.
Ref.4	Asignación de cursos, materias y cursos.
Ref.5	Emisión de certificados y partes.

BIBLIOTECA / INVENTARIO

REFERENCIA	FUNCIÓN
Ref.1	Alta – Modificación de libros.
Ref.2	Registro de prestación de libros.

Desarrollo de la propuesta

Para el desarrollo de esta investigación he utilizado diversas herramientas que me permitieron desarrollar de esta manera un Diseño Eficiente y Eficaz que contribuirá en el desempeño de las actividades académicas – administrativas de las Instituciones Educativas de Nivel Medio. Es así que para el desarrollo del modelo conceptual del sistema se utiliza Power Designer y de esta manera obtener el Script de la Base de Datos, la misma que se la realizó en Microsoft SQL Server.

FASE DE DISEÑO

En esta fase podemos detallar los procesos reales necesarios para el diseño del sistema escolástico.

PROCESOS GENERALES EN FUNCIÓN AL DISEÑO

REFERENCIA	FUNCIÓN
Ref.1	Registro del alumnado
Ref.2	Registro de personal docente
Ref.3	Registro de personal administrativo
Ref.4	Registro de planes de estudio
Ref. 5	Registro de materias
Ref. 6	Registro de asistencia
Ref. 7	Reportes de conducta
Ref. 8	Reportes de notas
Ref. 9	Reportes de asistencia
Ref. 10	Búsqueda de alumnos
Ref. 11	Certificados analíticos
Ref. 12	Permisos de pases

REGISTRO DE ALUMNOS

REFERENCIA	FUNCIÓN
Ref. 1	Presentación De certificados
Ref. 2	Certificado analítico
Ref. 3	Grafico de estado de materias
Ref. 4	Anotaciones particulares
Ref. 5	Auditoria de legajo
Ref. 6	Transacciones que involucran al alumno
Ref. 7	Control de exámenes
Ref. 8	Contactos
Ref. 9	Familiares o tutor

REGISTRO DE PERSONAL

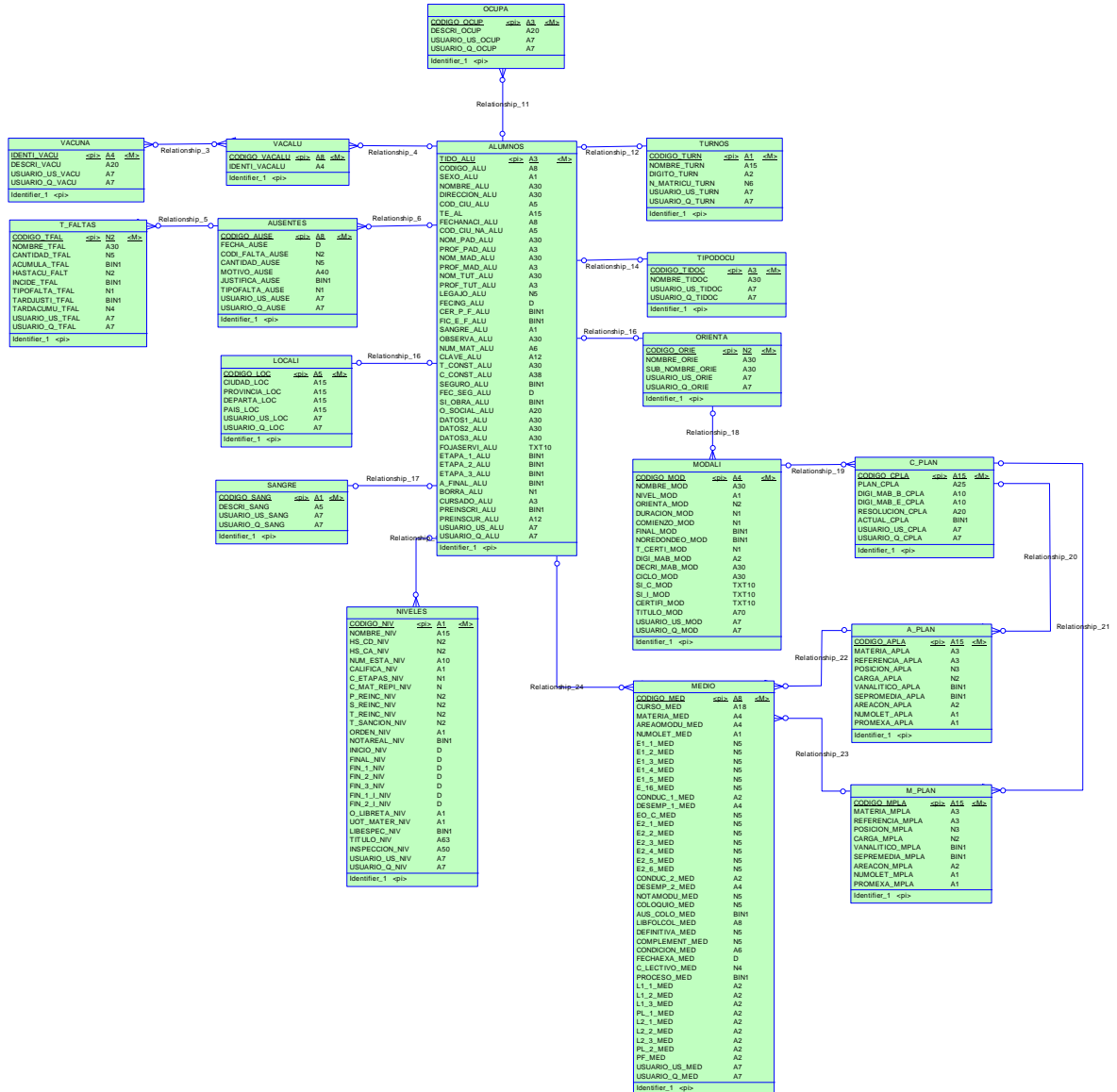
REFERENCIA	FUNCIÓN
Ref.1	Asignación de cargos
Ref.2	Asignación de materias
Ref.3	Asignación de cursos
Ref.4	Control de asistencia
Ref.5	Partes diarios de novedades
Ref. 6	Registración de títulos obtenidos

REGISTRO BIBLIOTECA/INVENTARIO

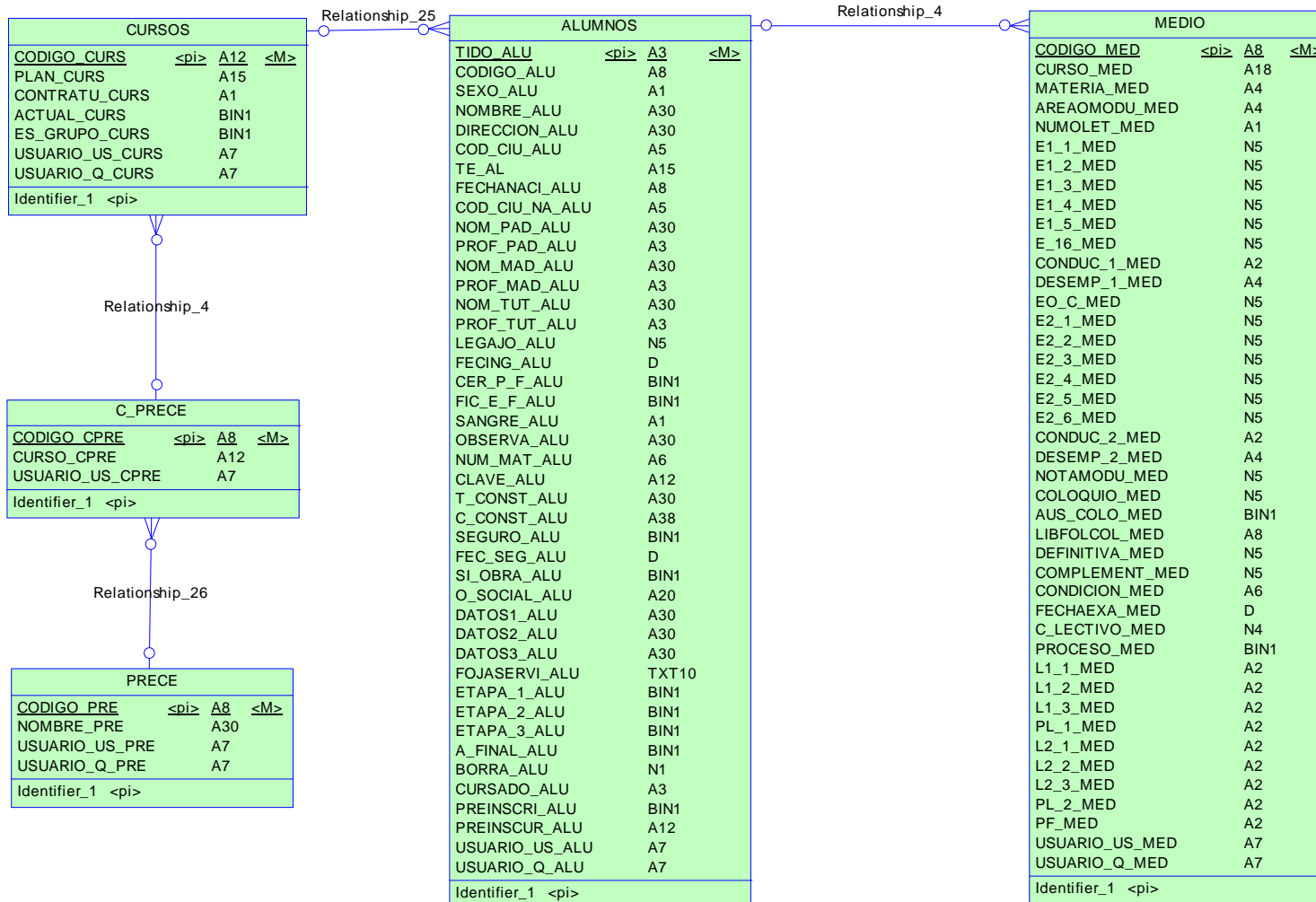
REFERENCIA	FUNCIÓN
Ref.1	Registración de títulos
Ref.2	Registro de prestación

MODELO CONCEPTUAL DE DATOS

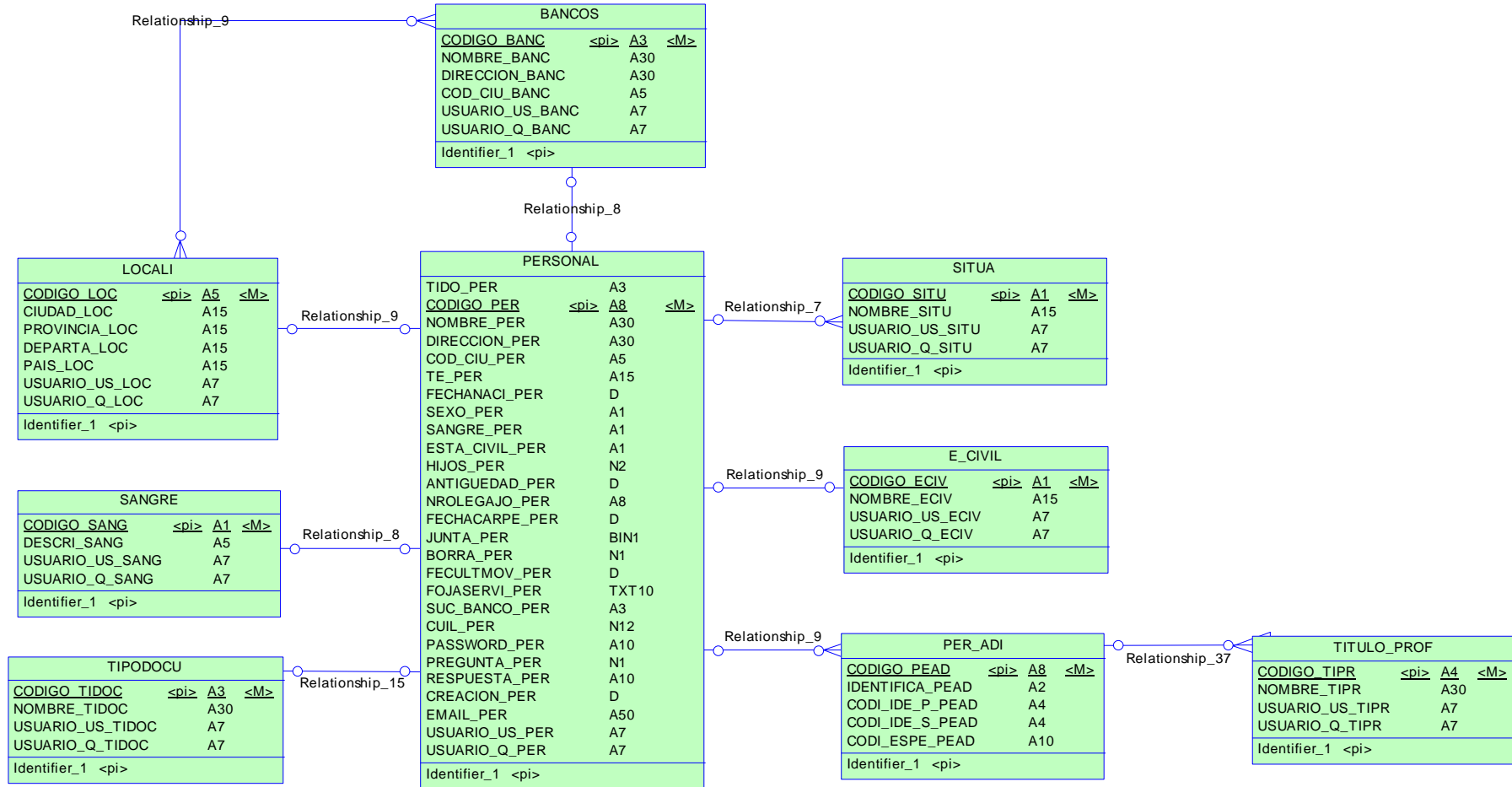
ALTA – MODIFICACIÓN DE ALUMNOS



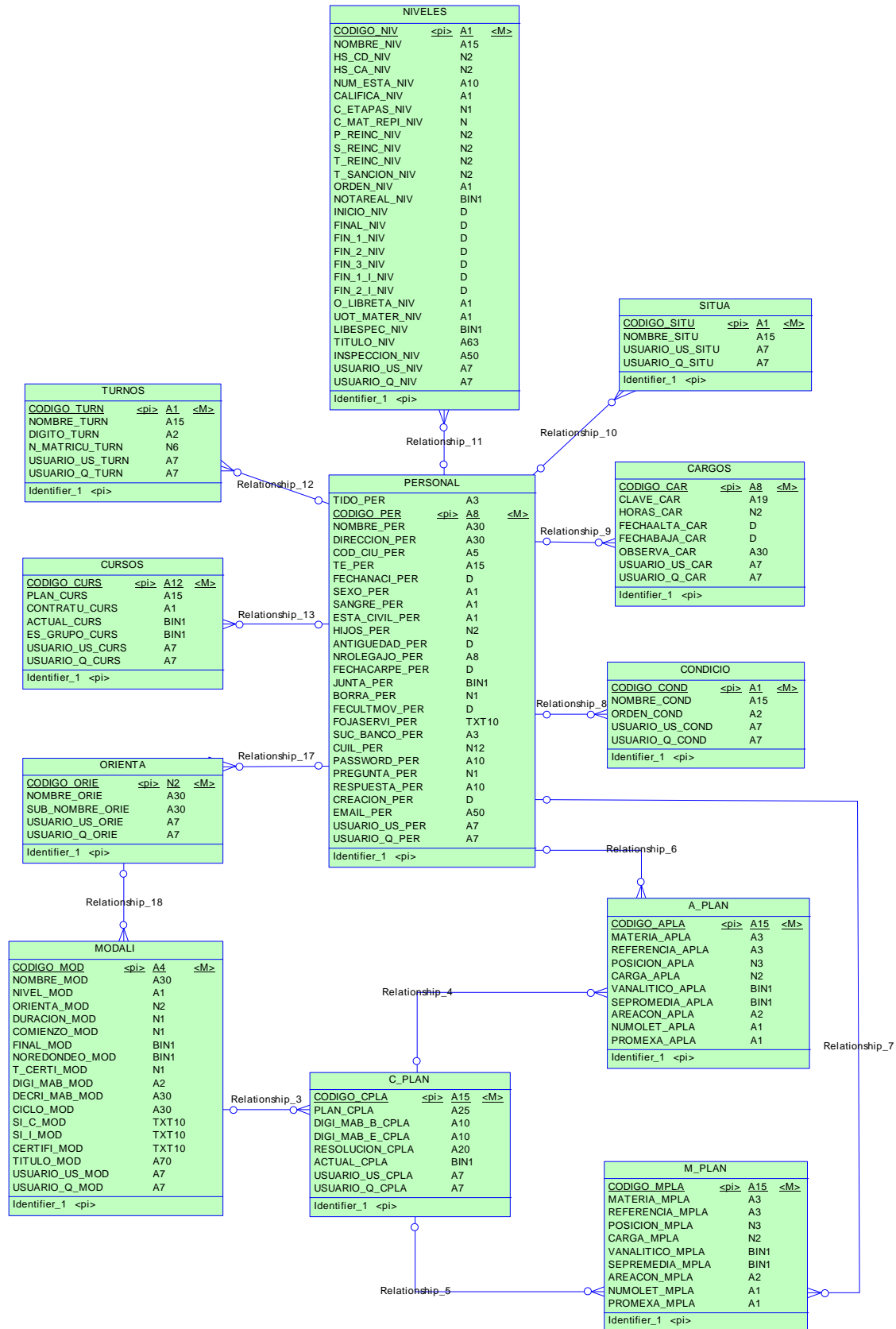
INGRESO DE NOTAS



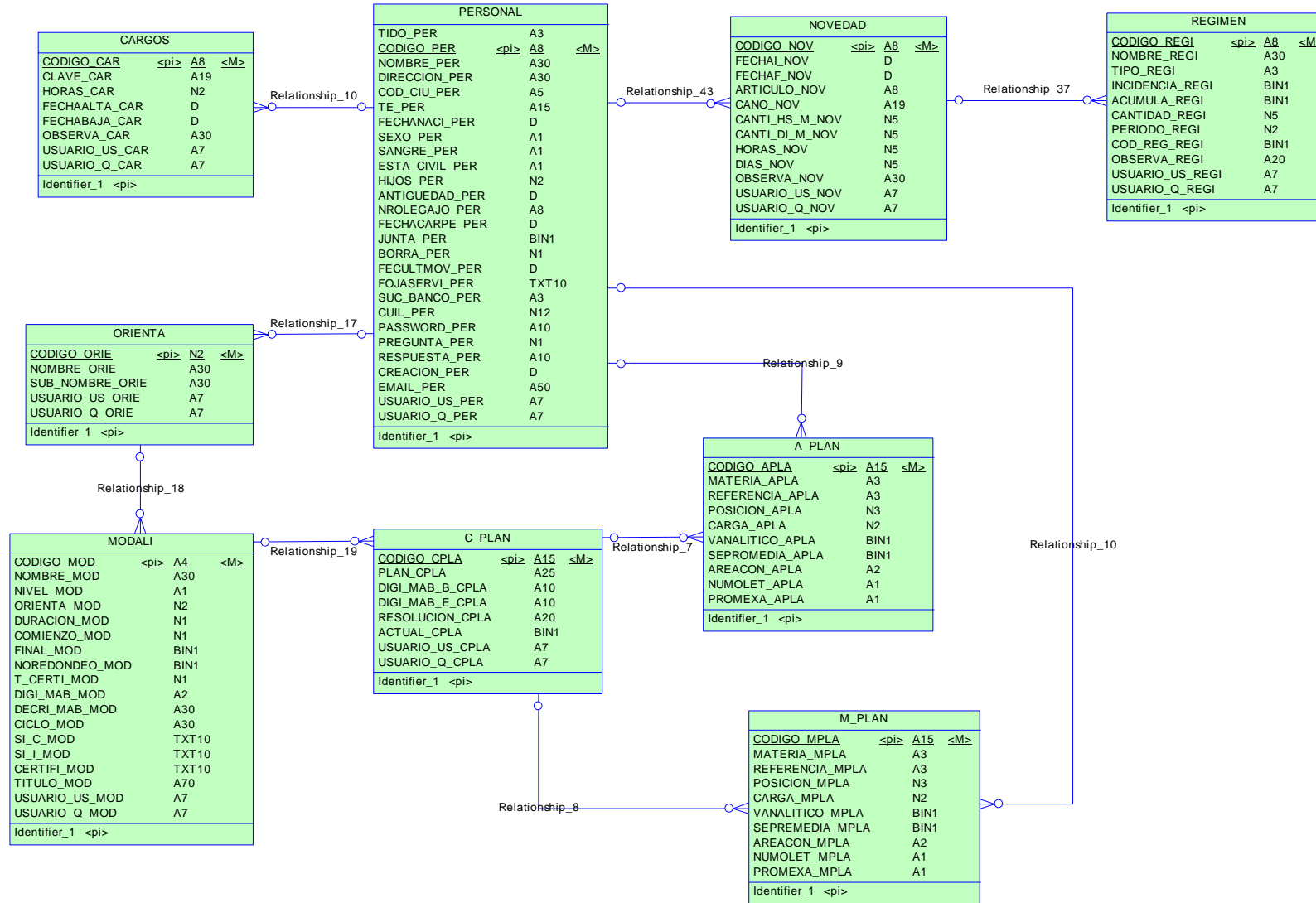
ALTA – MODIFICACIÓN DE PERSONAL



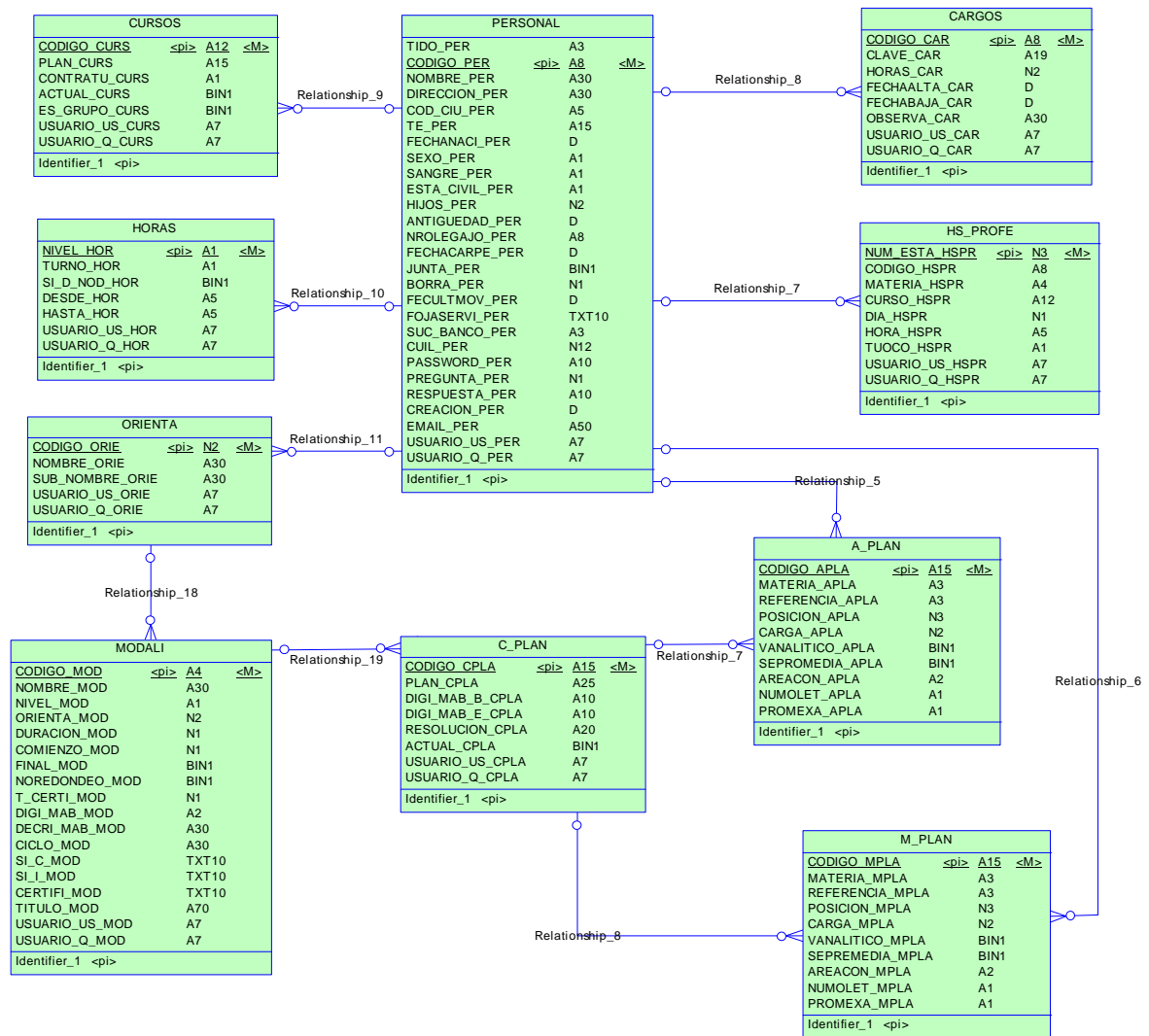
ASIGNACIÓN DE CARGOS



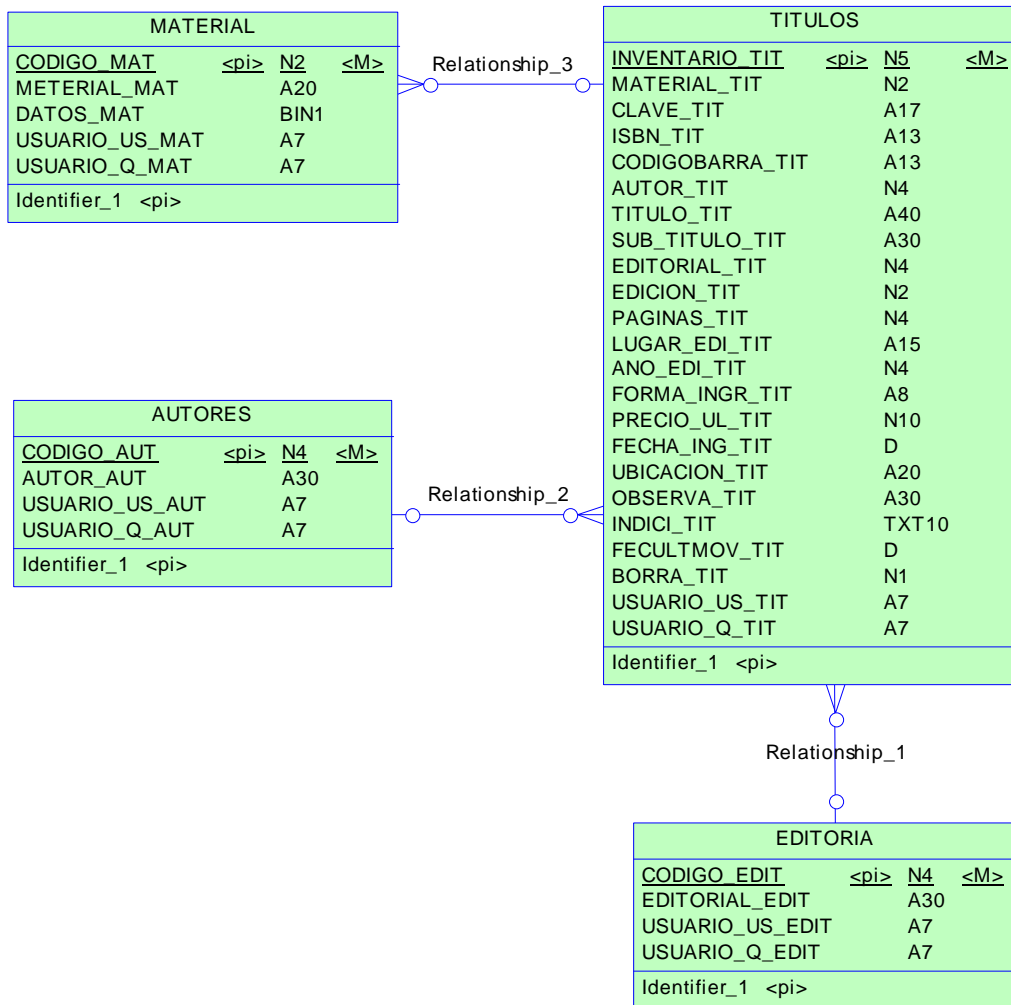
CONTROL DE AUSENTISMO



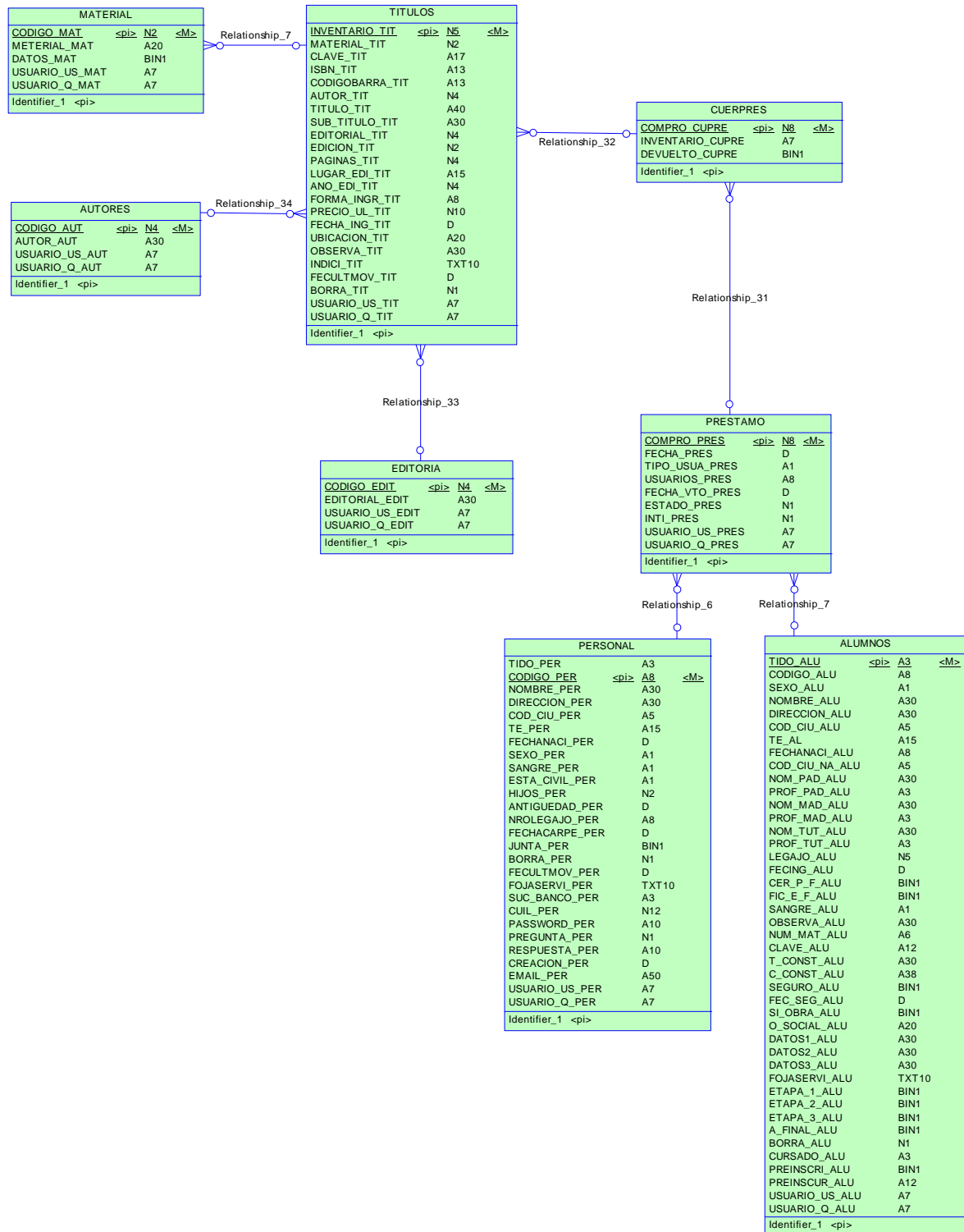
ASIGNACIÓN DE HORARIOS



BIBLIOTECA – INVENTARIOS

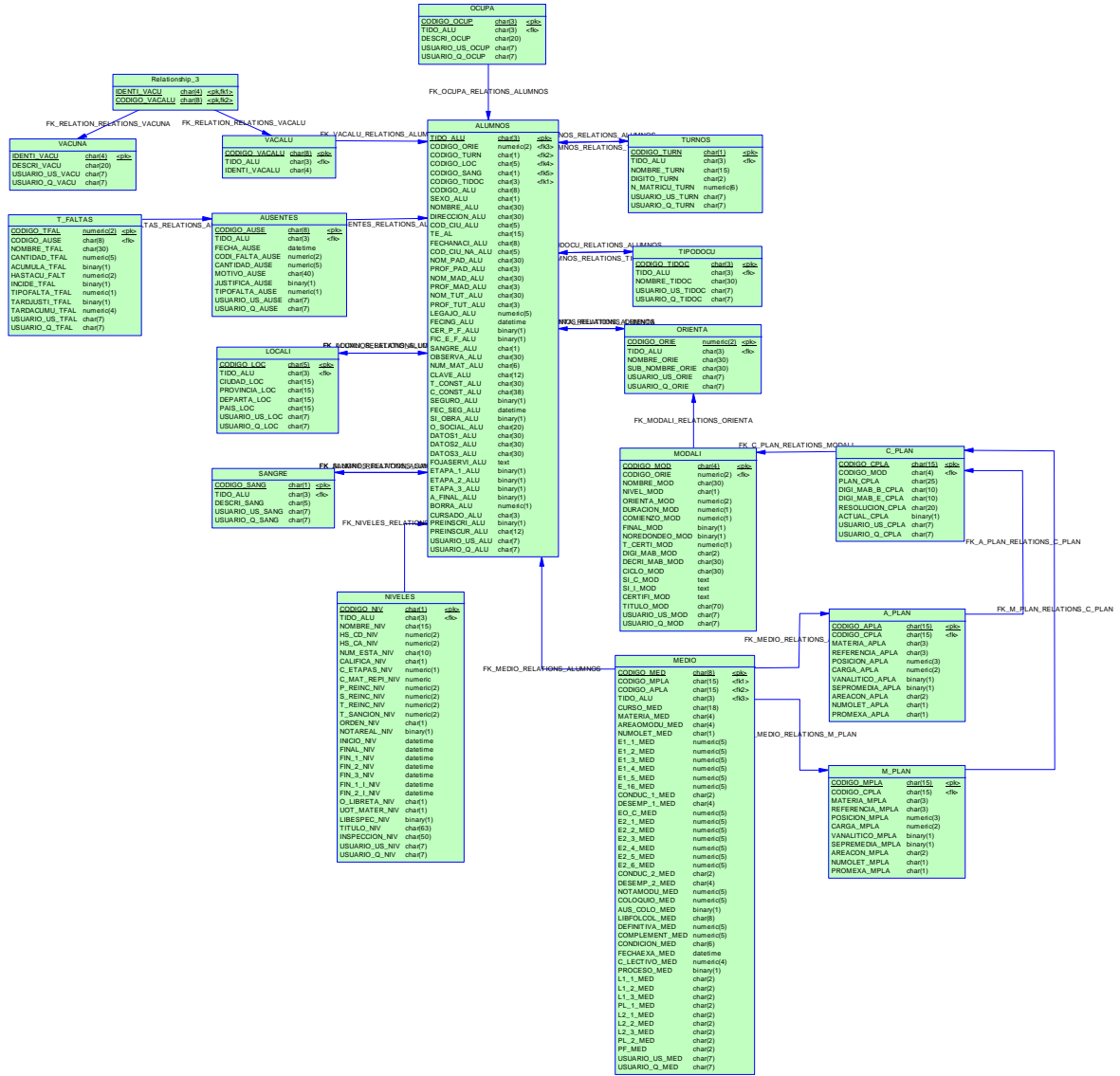


PRESTAMOS DE BIBLIOTECA

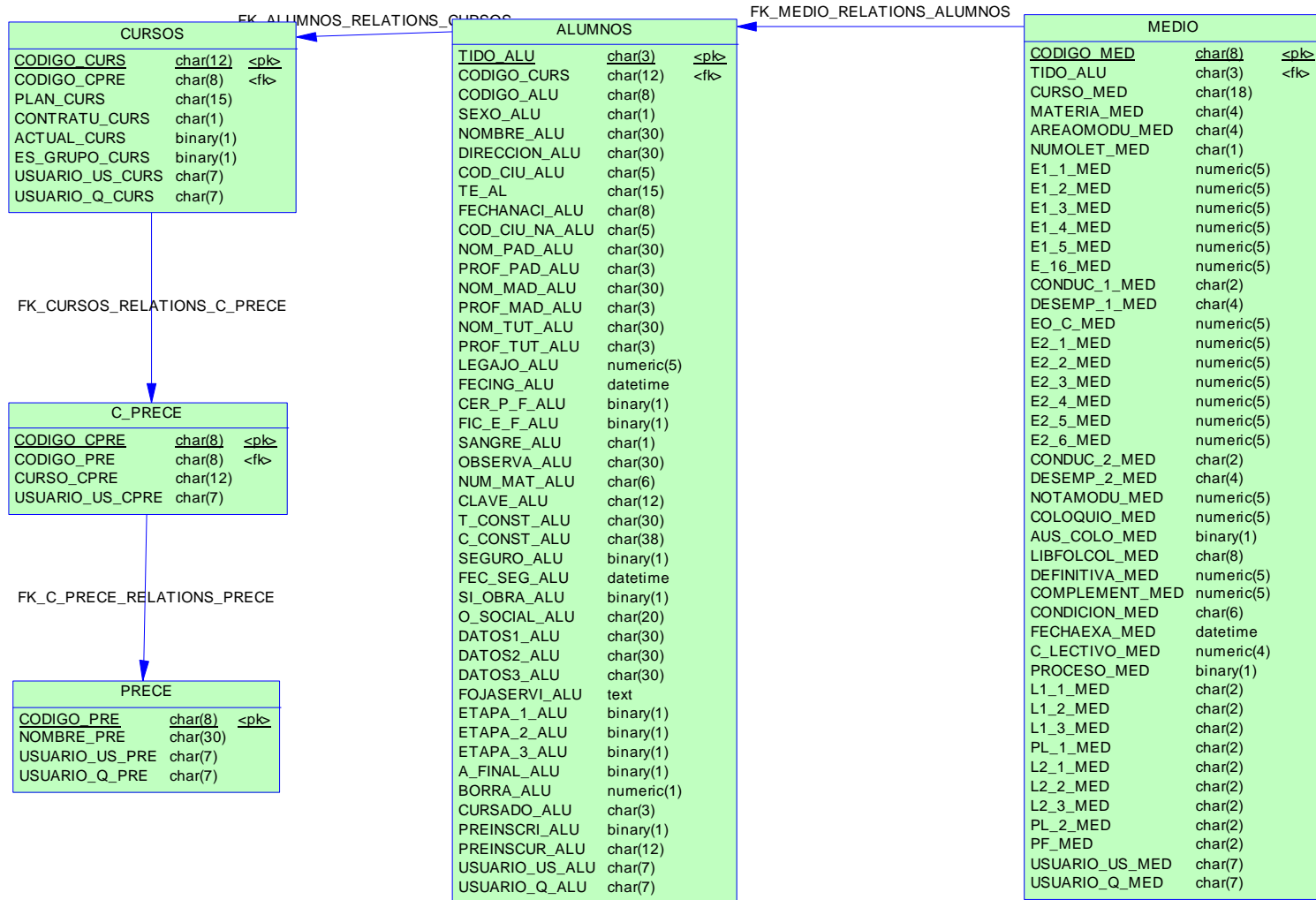


MODELO FÍSICO DE DATOS

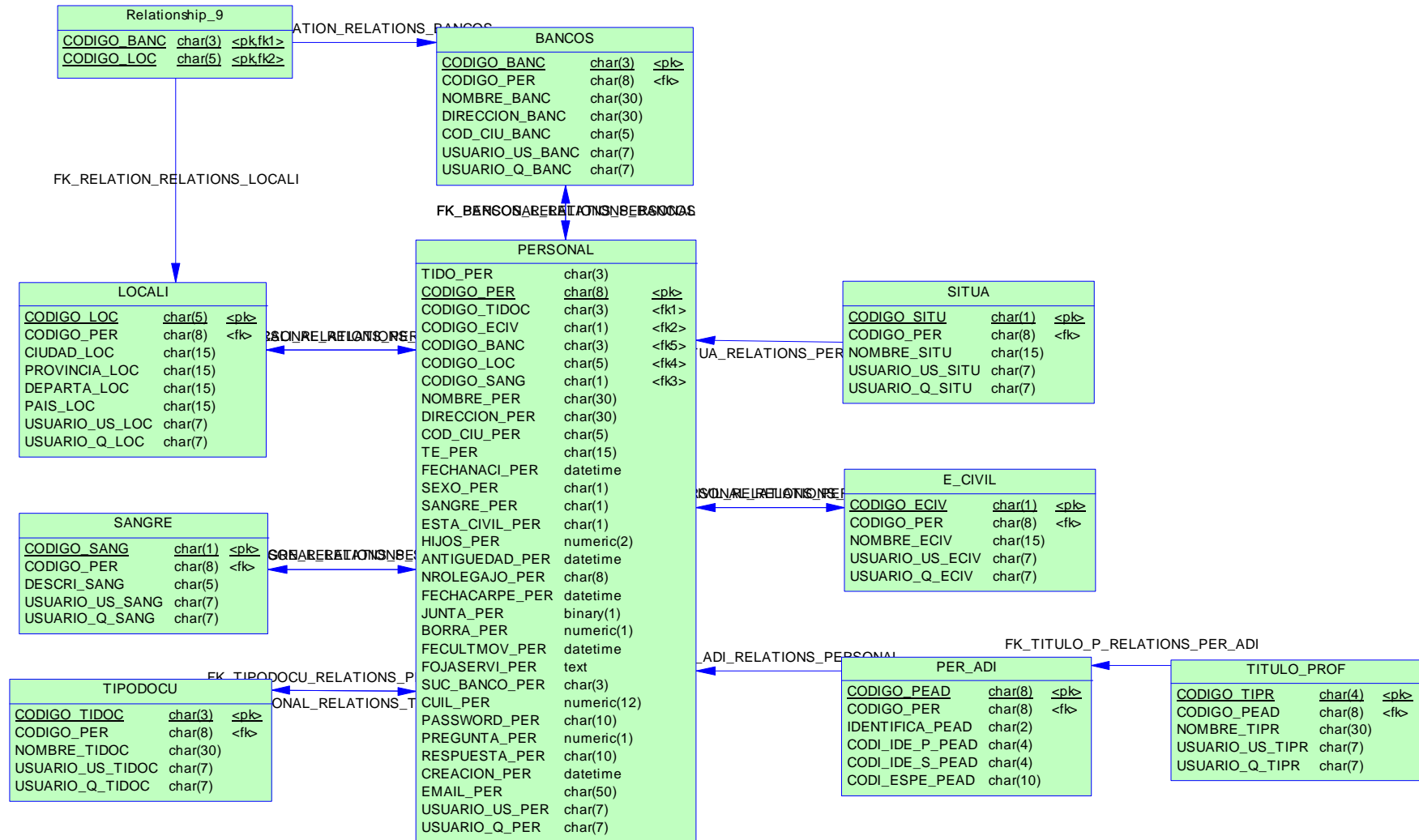
ALTA – MODIFICACIÓN DE ALUMNOS



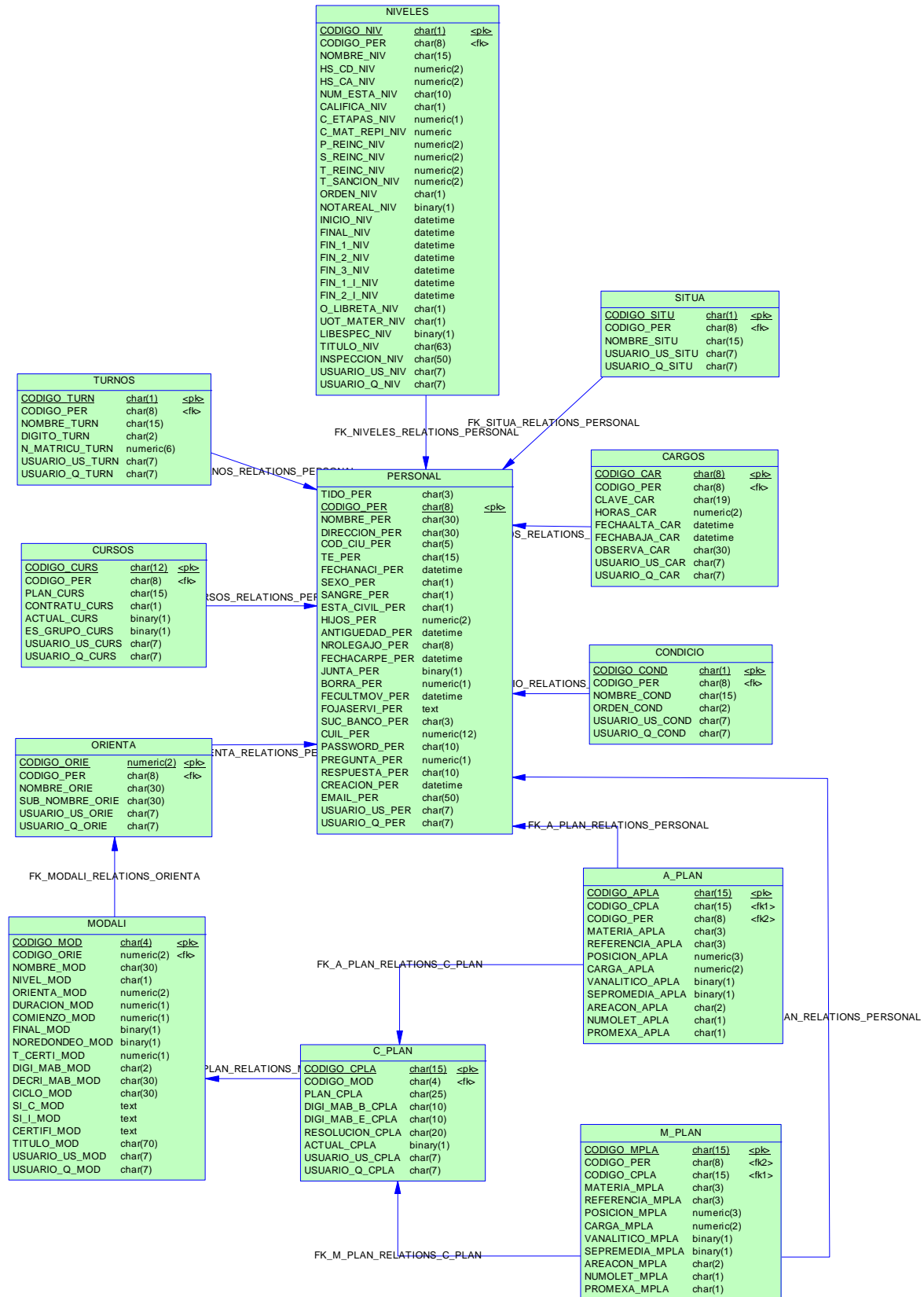
INGRESO DE NOTAS



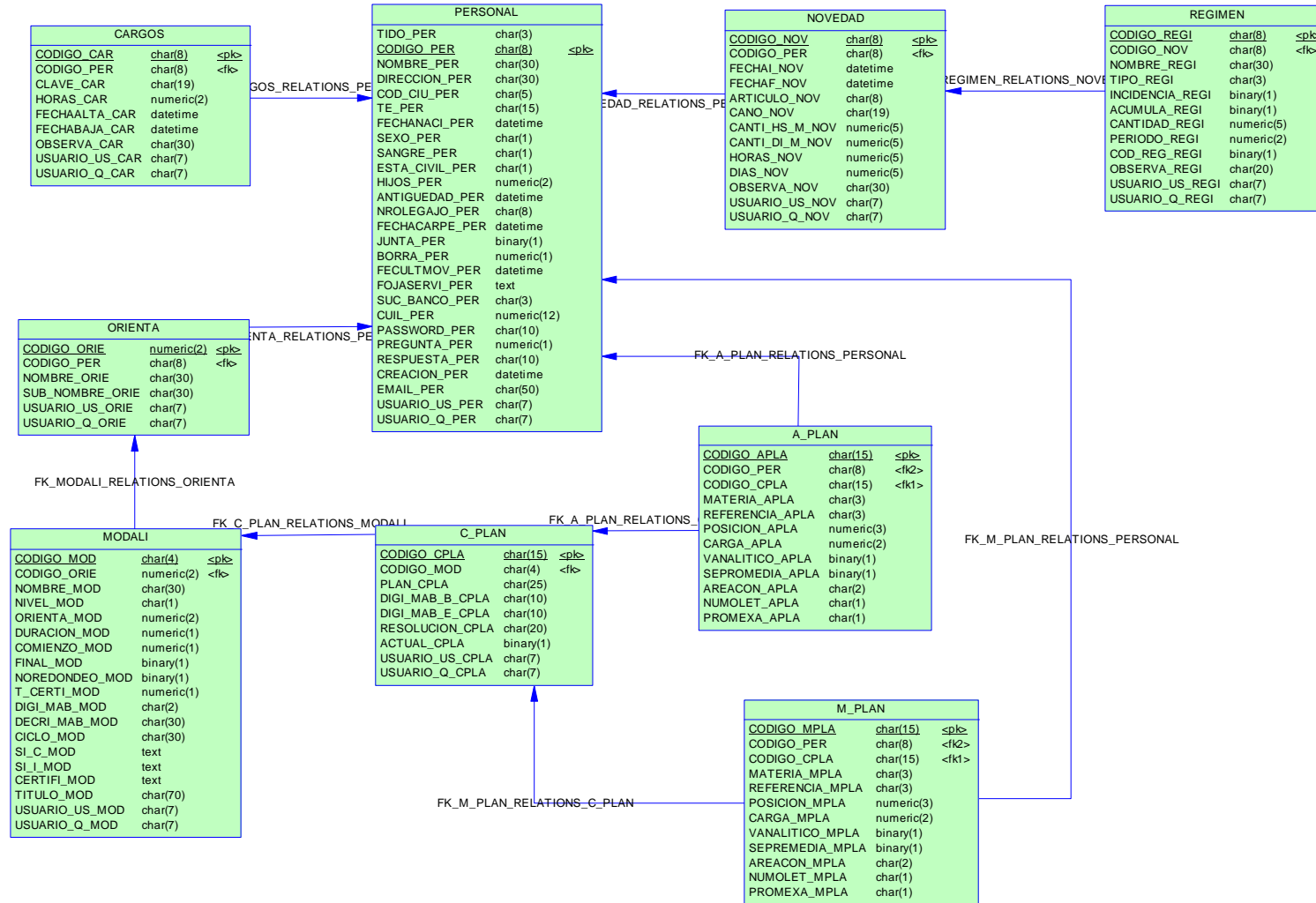
ALTA – MODIFICACIÓN DE PERSONAL



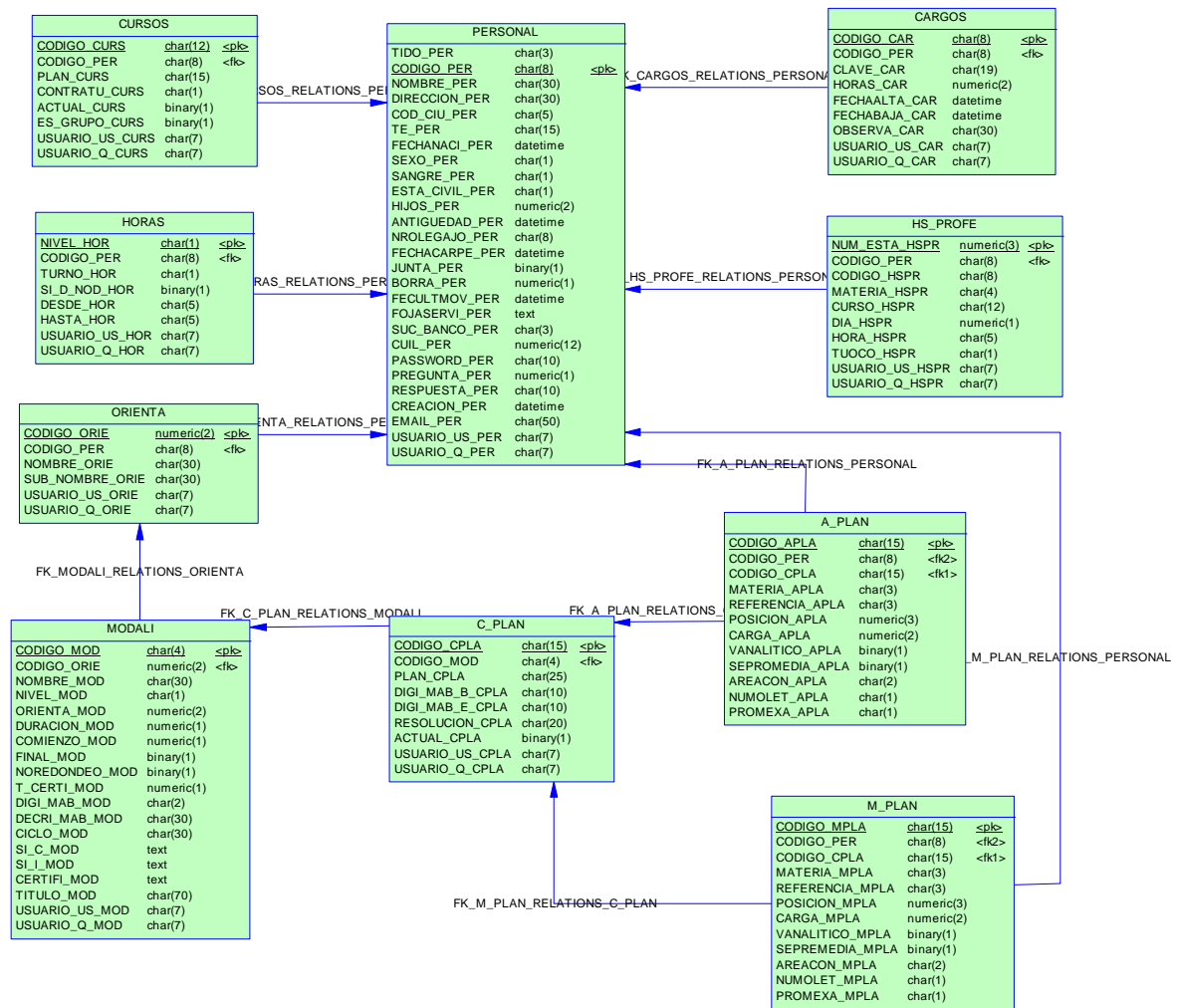
ASIGNACIÓN DE CARGOS



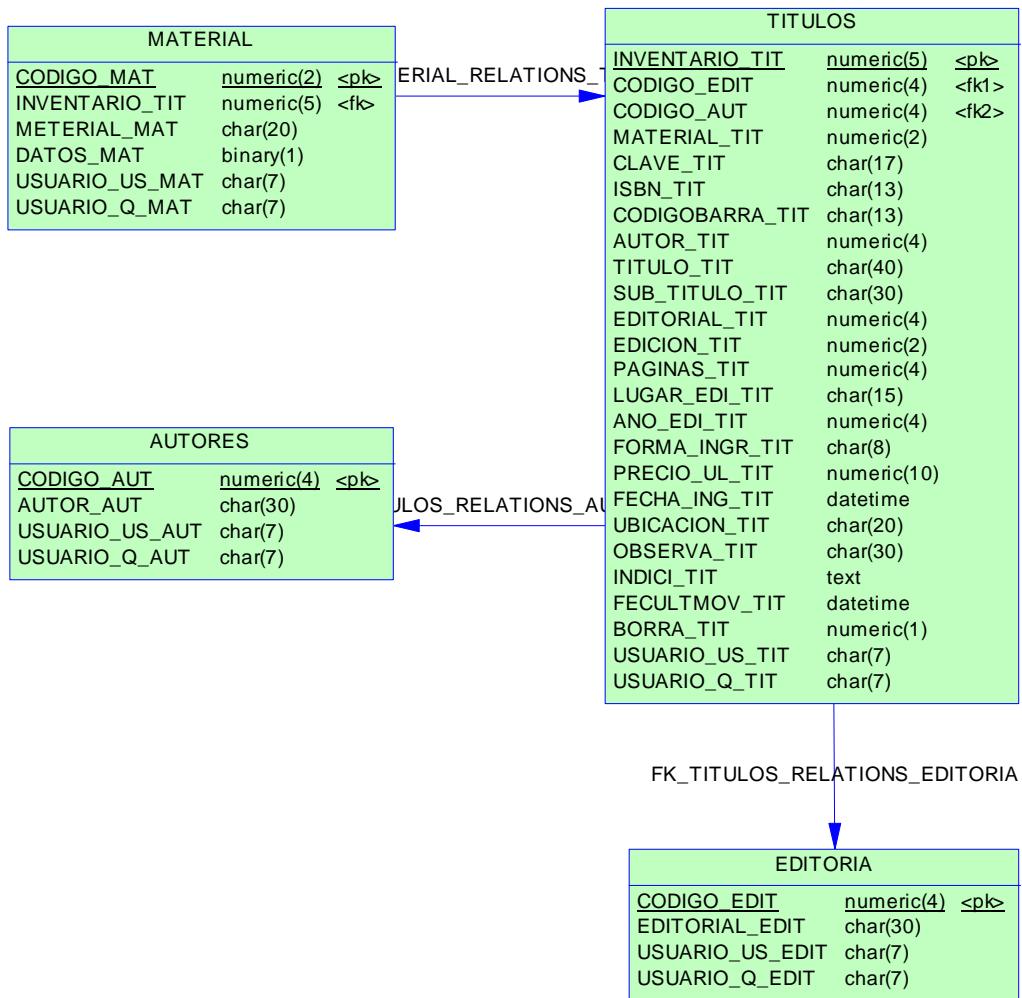
CONTROL DE AUSENTISMO



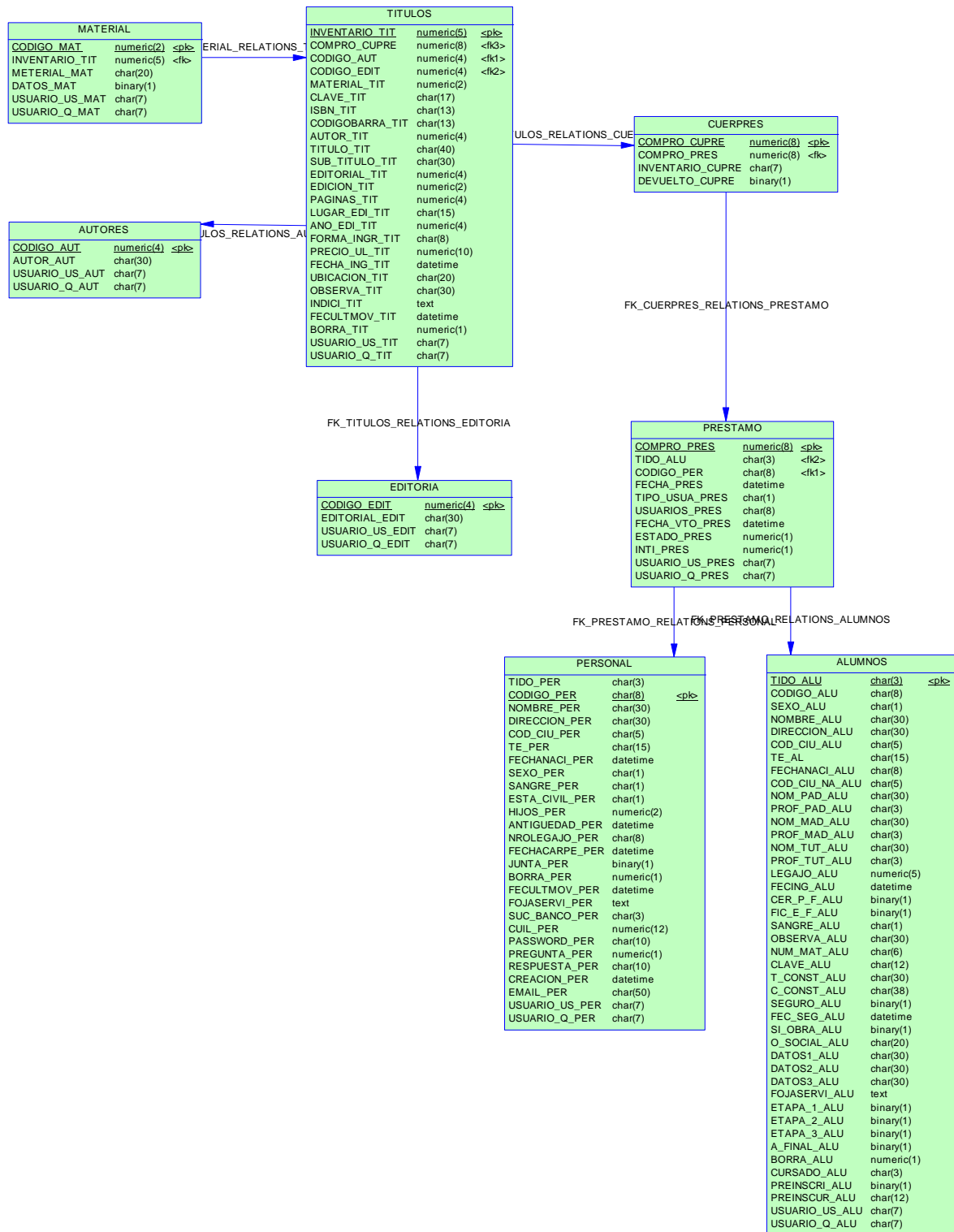
ASIGNACIÓN DE HORARIOS



BIBLIOTECA- INVENTARIO



PRESTAMOS DE BIBLIOTECA



Script para la creación de la Base de Datos

En el Script para la creación de la Base de datos se detallan en código todas las tablas que se utilizan en el Diseño del Modelo Conceptual.

Diseño de la Base Datos en SQL Server 2005

Se muestra todas las tablas que se utilizan para la elaboración del sistema, se especifican los atributos que cada uno de estas poseen.

CAPITULO IV

XIII. CONCLUSIONES

1. Las herramientas de diseño que son usadas en la actualidad nos ayudan en la corrección de errores antes de ejecutar el script de la base de datos.
2. El presente diseño ayuda en el manejo de la información para los procesos académicos y administrativos de las Instituciones de Educación Media, ya que agiliza dichos procesos intercambiando el trabajo manual por el sistema automatizado.
3. De igual forma la línea base del presente diseño puede ser aplicado en otras instituciones que ofertan otros servicios ya que se utilizan las mismas herramientas.

XIV. RECOMENDACIONES

1. Es necesario seguir los pasos establecidos para llegar a una buena documentación, y por ende a una buena base de datos depurada y exenta de errores.
2. Las instituciones deberían automatizar sus procesos para poder manejar de una manera optima sus flujos de información.
3. Recomiendo utilizar el programa Power Designer como herramienta modeladora de datos ya que nos permite revisar en todo momento el flujo de información del modelo entidad relación desarrollado.

XV. BIBLIOGRAFÍA

Bibliografía Consultada

- <http://www.monografias.com/trabajos5/edenarg/edenarg.shtml?relacionados>
- http://www.oei.org.ar/edumedia/pdfs/T07_Docu1_Laeducacionmedia_Pereyra_Batuk_Inigo_Izcovich%20.pdf
- http://www.capitannemo.com.ar/de_la_educacion_media_o_la_media.htm
- <http://www.cengage.com.ar/educacion-media/>
- http://www.smiba.org.ar/med_interna/vol_05/05_01.htm
- <http://www.sybase.com.ar/products/modelingmetadata/powerdesigner.shtml>
- <http://es.wikipedia.org/wiki/MicrosoftSqlServer>
- http://es.wikipedia.org/wiki/Base_de_Datos
- http://www.bn.gov.ar/archivos/anexos_proyectos_especiales/encuentro/ponencias/ponencia_Patton_castellano.pdf
- http://www.frlp.utn.edu.ar/web/ceneisi/modelo_conceptual_gramatica.pdf
- http://www.graduadosportuaria.com.ar/Vias%20Navegables/Tema%2021_Modelos%20fisicos.pdf
- <http://ict.udlap.mx/people/carlos/is341/bases07.html>
- <http://www.willydev.net/InsiteCreation/v1.0/descargas/prev/explicaxp.pdfg>
- <http://metodologiasagilesyagentes.blogspot.com/2008/04/programacin-extrema-xp.html>
- <http://www.mtbase.com/productos/modelamientometadatos/powerdesigner>
- <http://www.dlsi.ua.es/asignaturas/dpaa/tema1.pdf>
- <http://www.programacionextrema.org/>
- <http://www.microsoft.com/latam/sql/>

ANEXOS

SCRIPT PARA LA CREACION DE LA BASE DE DATOS

```

/*=====*/
/* Table: ALUMNOS                                     */
/*=====*/
create table ALUMNOS (
  TIDO_ALU      char(3)      not null,
  CODIGO_TURN   char(1)      null,
  CODIGO_ORIE   numeric(2)   null,
  CODIGO_TIDOC  char(3)      null,
  CODIGO_CURS   char(12)     null,
  CODIGO_SANG   char(1)      null,
  CODIGO_LOC    char(5)      null,
  CODIGO_ALU    char(8)      null,
  SEXO_ALU      char(1)      null,
  NOMBRE_ALU    char(30)     null,
  DIRECCION_ALU char(30)     null,
  COD_CIU_ALU   char(5)      null,
  TE_AL         char(15)     null,
  FECHANACI_ALU char(8)      null,
  COD_CIU_NA_ALU char(5)     null,
  NOM_PAD_ALU   char(30)     null,
  PROF_PAD_ALU  char(3)      null,
  NOM_MAD_ALU   char(30)     null,
  PROF_MAD_ALU  char(3)      null,
  NOM_TUT_ALU   char(30)     null,
  PROF_TUT_ALU  char(3)      null,
  LEGAJO_ALU    numeric(5)   null,
  FECING_ALU    datetime     null,
  CER_P_F_ALU   binary(1)    null,
  FIC_E_F_ALU   binary(1)    null,
  SANGRE_ALU    char(1)      null,
  OBSERVA_ALU   char(30)     null,
  NUM_MAT_ALU   char(6)      null,
  CLAVE_ALU     char(12)     null,
  T_CONST_ALU   char(30)     null,
  C_CONST_ALU   char(38)     null,
  SEGURO_ALU    binary(1)    null,
  FEC_SEG_ALU   datetime     null,
  SI_OBRA_ALU   binary(1)    null,
  O_SOCIAL_ALU  char(20)     null,
  DATOS1_ALU    char(30)     null,
  DATOS2_ALU    char(30)     null,
  DATOS3_ALU    char(30)     null,
  FOJASERVI_ALU text         null,
  ETAPA_1_ALU   binary(1)    null,
  ETAPA_2_ALU   binary(1)    null,
  ETAPA_3_ALU   binary(1)    null,
  A_FINAL_ALU   binary(1)    null,
  BORRA_ALU     numeric(1)    null,
  CURSADO_ALU   char(3)      null,
  PREINSCRI_ALU binary(1)    null,
  PREINSCUR_ALU char(12)     null,
  USUARIO_US_ALU char(7)     null,
  USUARIO_Q_ALU char(7)     null,
  constraint PK_ALUMNOS primary key (TIDO_ALU)
)
go

```

```

/*=====*/
/* Index: RELATIONSHIP_14_FK          */
/*=====*/
create index RELATIONSHIP_14_FK on ALUMNOS (
CODIGO_TURN
)
go

```

```

/*=====*/
/* Index: RELATIONSHIP_17_FK          */
/*=====*/
create index RELATIONSHIP_17_FK on ALUMNOS (
CODIGO_TIDOC
)
go

```

```

/*=====*/
/* Index: RELATIONSHIP_21_FK          */
/*=====*/
create index RELATIONSHIP_21_FK on ALUMNOS (
CODIGO_ORIE
)
go

```

```

/*=====*/
/* Index: RELATIONSHIP_30_FK          */
/*=====*/
create index RELATIONSHIP_30_FK on ALUMNOS (
CODIGO_CURS
)
go

```

```

/*=====*/
/* Index: RELATIONSHIP_40_FK          */
/*=====*/
create index RELATIONSHIP_40_FK on ALUMNOS (
CODIGO_SANG
)
go

```

```

/*=====*/
/* Index: RELATIONSHIP_41_FK          */
/*=====*/
create index RELATIONSHIP_41_FK on ALUMNOS (
CODIGO_LOC
)
go

```

```

/*=====*/
/* Table: AUSENTES                    */
/*=====*/
create table AUSENTES (

```

```

CODIGO_AUSE      char(8)      not null,
TIDO_ALU         char(3)       null,
FECHA_AUSE       datetime     null,
CODI_FALTA_AUSE  numeric(2)    null,
CANTIDAD_AUSE    numeric(5)    null,
MOTIVO_AUSE      char(40)     null,
JUSTIFICA_AUSE   binary(1)    null,
TIPOFALTA_AUSE   numeric(1)    null,
USUARIO_US_AUSE  char(7)       null,
USUARIO_Q_AUSE   char(7)       null,
constraint PK_AUSENTES primary key (CODIGO_AUSE)
)
go

```

```

/*=====*/
/* Index: RELATIONSHIP_7_FK                */
/*=====*/
create index RELATIONSHIP_7_FK on AUSENTES (
TIDO_ALU
)
go

```

```

/*=====*/
/* Table: AUTORES                          */
/*=====*/
create table AUTORES (
CODIGO_AUT      numeric(4)    not null,
AUTOR_AUT       char(30)     null,
USUARIO_US_AUT  char(7)       null,
USUARIO_Q_AUT   char(7)       null,
constraint PK_AUTORES primary key (CODIGO_AUT)
)
go

```

```

/*=====*/
/* Table: A_PLAN                            */
/*=====*/
create table A_PLAN (
CODIGO_APLA     char(15)     not null,
CODIGO_PER      char(8)      null,
CODIGO_CPLA     char(15)     null,
MATERIA_APLA    char(3)      null,
REFERENCIA_APLA char(3)      null,
POSICION_APLA   numeric(3)   null,
CARGA_APLA      numeric(2)   null,
VANALITICO_APLA binary(1)   null,
SEPMEDIA_APLA   binary(1)   null,
AREACON_APLA    char(2)      null,
NUMOLET_APLA    char(1)      null,
PROMEXA_APLA    char(1)      null,
constraint PK_A_PLAN primary key (CODIGO_APLA)
)
go

```

```

/*=====*/
/* Index: RELATIONSHIP_62_FK */
/*=====*/
create index RELATIONSHIP_62_FK on A_PLAN (
CODIGO_CPLA
)
go

```

```

/*=====*/
/* Index: RELATIONSHIP_63_FK */
/*=====*/
create index RELATIONSHIP_63_FK on A_PLAN (
CODIGO_PER
)
go

```

```

/*=====*/
/* Table: BANCOS */
/*=====*/
create table BANCOS (
CODIGO_BANC char(3) not null,
CODIGO_PER char(8) null,
NOMBRE_BANC char(30) null,
DIRECCION_BANC char(30) null,
COD_CIU_BANC char(5) null,
USUARIO_US_BANC char(7) null,
USUARIO_Q_BANC char(7) null,
constraint PK_BANCOS primary key (CODIGO_BANC)
)
go

```

```

/*=====*/
/* Index: RELATIONSHIP_51_FK */
/*=====*/
create index RELATIONSHIP_51_FK on BANCOS (
CODIGO_PER
)
go

```

```

/*=====*/
/* Table: CARGOS */
/*=====*/
create table CARGOS (
CODIGO_CAR char(8) not null,
CODIGO_PER char(8) null,
CLAVE_CAR char(19) null,
HORAS_CAR numeric(2) null,
FECHAALTA_CAR datetime null,
FECHABAJA_CAR datetime null,
OBSERVA_CAR char(30) null,
USUARIO_US_CAR char(7) null,
USUARIO_Q_CAR char(7) null,
constraint PK_CARGOS primary key (CODIGO_CAR)
)

```

go

```
/*=====*/
/* Index: RELATIONSHIP_52_FK */
/*=====*/
create index RELATIONSHIP_52_FK on CARGOS (
CODIGO_PER
)
go
```

```
/*=====*/
/* Table: CONDICIO */
/*=====*/
create table CONDICIO (
CODIGO_COND char(1) not null,
CODIGO_PER char(8) null,
NOMBRE_COND char(15) null,
ORDEN_COND char(2) null,
USUARIO_US_COND char(7) null,
USUARIO_Q_COND char(7) null,
constraint PK_CONDICIO primary key (CODIGO_COND)
)
go
```

```
/*=====*/
/* Index: RELATIONSHIP_54_FK */
/*=====*/
create index RELATIONSHIP_54_FK on CONDICIO (
CODIGO_PER
)
go
```

```
/*=====*/
/* Table: CUERPRES */
/*=====*/
create table CUERPRES (
COMPRO_CUPRE numeric(8) not null,
COMPRO_PRES numeric(8) null,
INVENTARIO_CUPRE char(7) null,
DEVUELTO_CUPRE binary(1) null,
constraint PK_CUERPRES primary key (COMPRO_CUPRE)
)
go
```

```
/*=====*/
/* Index: RELATIONSHIP_35_FK */
/*=====*/
create index RELATIONSHIP_35_FK on CUERPRES (
COMPRO_PRES
)
go
```

```

/*=====*/
/* Table: CURSOS */
/*=====*/
create table CURSOS (
  CODIGO_CURS      char(12)      not null,
  CODIGO_CPRES     char(8)       null,
  TIDO_ALU         char(3)       null,
  CODIGO_PER       char(8)       null,
  PLAN_CURS       char(15)      null,
  CONTRATU_CURS   char(1)       null,
  ACTUAL_CURS     binary(1)     null,
  ES_GRUPO_CURS   binary(1)     null,
  USUARIO_US_CURS char(7)       null,
  USUARIO_Q_CURS  char(7)       null,
  constraint PK_CURSOS primary key (CODIGO_CURS)
)
go

```

```

/*=====*/
/* Index: RELATIONSHIP_31_FK */
/*=====*/
create index RELATIONSHIP_31_FK on CURSOS (
  TIDO_ALU
)
go

```

```

/*=====*/
/* Index: RELATIONSHIP_59_FK */
/*=====*/
create index RELATIONSHIP_59_FK on CURSOS (
  CODIGO_CPRES
)
go

```

```

/*=====*/
/* Index: RELATIONSHIP_60_FK */
/*=====*/
create index RELATIONSHIP_60_FK on CURSOS (
  CODIGO_PER
)
go

```

```

/*=====*/
/* Table: C_PLAN */
/*=====*/
create table C_PLAN (
  CODIGO_CPLA     char(15)      not null,
  CODIGO_MOD      char(4)       null,
  PLAN_CPLA       char(25)     null,
  DIGI_MAB_B_CPLA char(10)     null,
  DIGI_MAB_E_CPLA char(10)     null,
  RESOLUCION_CPLA char(20)     null,
  ACTUAL_CPLA     binary(1)    null,
  USUARIO_US_CPLA char(7)      null,
)

```

```

USUARIO_Q_CPLA char(7) null,
constraint PK_C_PLAN primary key (CODIGO_CPLA)
)
go

```

```

/*=====*/
/* Index: RELATIONSHIP_25_FK */
/*=====*/
create index RELATIONSHIP_25_FK on C_PLAN (
CODIGO_MOD
)
go

```

```

/*=====*/
/* Table: C_PRECE */
/*=====*/
create table C_PRECE (
CODIGO_CPRE char(8) not null,
CODIGO_PRE char(8) null,
CURSO_CPRE char(12) null,
USUARIO_US_CPRE char(7) null,
constraint PK_C_PRECE primary key (CODIGO_CPRE)
)
go

```

```

/*=====*/
/* Index: RELATIONSHIP_32_FK */
/*=====*/
create index RELATIONSHIP_32_FK on C_PRECE (
CODIGO_PRE
)
go

```

```

/*=====*/
/* Table: EDITORIA */
/*=====*/
create table EDITORIA (
CODIGO_EDIT numeric(4) not null,
EDITORIAL_EDIT char(30) null,
USUARIO_US_EDIT char(7) null,
USUARIO_Q_EDIT char(7) null,
constraint PK_EDITORIA primary key (CODIGO_EDIT)
)
go

```

```

/*=====*/
/* Table: E_CIVIL */
/*=====*/
create table E_CIVIL (
CODIGO_ECIV char(1) not null,
CODIGO_PER char(8) null,
NOMBRE_ECIV char(15) null,
USUARIO_US_ECIV char(7) null,

```

```

    USUARIO_Q_ECIV    char(7)    null,
    constraint PK_E_CIVIL primary key (CODIGO_ECIV)
)
go

```

```

/*=====*/
/* Index: RELATIONSHIP_49_FK                               */
/*=====*/
create index RELATIONSHIP_49_FK on E_CIVIL (
CODIGO_PER
)
go

```

```

/*=====*/
/* Table: HORAS                                           */
/*=====*/
create table HORAS (
    NIVEL_HOR    char(1)    not null,
    CODIGO_PER    char(8)    null,
    TURNO_HOR    char(1)    null,
    SI_D_NOD_HOR    binary(1)    null,
    DESDE_HOR    char(5)    null,
    HASTA_HOR    char(5)    null,
    USUARIO_US_HOR    char(7)    null,
    USUARIO_Q_HOR    char(7)    null,
    constraint PK_HORAS primary key (NIVEL_HOR)
)
go

```

```

/*=====*/
/* Index: RELATIONSHIP_56_FK                               */
/*=====*/
create index RELATIONSHIP_56_FK on HORAS (
CODIGO_PER
)
go

```

```

/*=====*/
/* Table: HS_PROFE                                         */
/*=====*/
create table HS_PROFE (
    NUM_ESTA_HSPR    numeric(3)    not null,
    CODIGO_PER    char(8)    null,
    CODIGO_HSPR    char(8)    null,
    MATERIA_HSPR    char(4)    null,
    CURSO_HSPR    char(12)    null,
    DIA_HSPR    numeric(1)    null,
    HORA_HSPR    char(5)    null,
    TUOCO_HSPR    char(1)    null,
    USUARIO_US_HSPR    char(7)    null,
    USUARIO_Q_HSPR    char(7)    null,
    constraint PK_HS_PROFE primary key (NUM_ESTA_HSPR)
)
go

```

```

/*=====*/
/* Index: RELATIONSHIP_55_FK */
/*=====*/
create index RELATIONSHIP_55_FK on HS_PROFE (
CODIGO_PER
)
go

```

```

/*=====*/
/* Table: LOCALI */
/*=====*/
create table LOCALI (
CODIGO_LOC char(5) not null,
CIUDAD_LOC char(15) null,
PROVINCIA_LOC char(15) null,
DEPARTA_LOC char(15) null,
PAIS_LOC char(15) null,
USUARIO_US_LOC char(7) null,
USUARIO_Q_LOC char(7) null,
constraint PK_LOCALI primary key (CODIGO_LOC)
)
go

```

```

/*=====*/
/* Table: MATERIAL */
/*=====*/
create table MATERIAL (
CODIGO_MAT numeric(2) not null,
INVENTARIO_TIT numeric(5) null,
METERIAL_MAT char(20) null,
DATOS_MAT binary(1) null,
USUARIO_US_MAT char(7) null,
USUARIO_Q_MAT char(7) null,
constraint PK_MATERIAL primary key (CODIGO_MAT)
)
go

```

```

/*=====*/
/* Index: RELATIONSHIP_39_FK */
/*=====*/
create index RELATIONSHIP_39_FK on MATERIAL (
INVENTARIO_TIT
)
go

```

```

/*=====*/
/* Table: MEDIO */
/*=====*/
create table MEDIO (
CODIGO_MED char(8) not null,
CODIGO_MPLA char(15) null,
CODIGO_APLA char(15) null,

```

```

TIDO_ALU      char(3)      null,
CURSO_MED     char(18)     null,
MATERIA_MED   char(4)      null,
AREAOMODU_MED char(4)      null,
NUMOLET_MED   char(1)      null,
E1_1_MED     numeric(5)    null,
E1_2_MED     numeric(5)    null,
E1_3_MED     numeric(5)    null,
E1_4_MED     numeric(5)    null,
E1_5_MED     numeric(5)    null,
E_16_MED     numeric(5)    null,
CONDOC_1_MED  char(2)      null,
DESEMP_1_MED  char(4)      null,
EO_C_MED      numeric(5)    null,
E2_1_MED     numeric(5)    null,
E2_2_MED     numeric(5)    null,
E2_3_MED     numeric(5)    null,
E2_4_MED     numeric(5)    null,
E2_5_MED     numeric(5)    null,
E2_6_MED     numeric(5)    null,
CONDOC_2_MED  char(2)      null,
DESEMP_2_MED  char(4)      null,
NOTAMODU_MED  numeric(5)    null,
COLOQUIO_MED  numeric(5)    null,
AUS_COLO_MED  binary(1)     null,
LIBFOLCOL_MED char(8)      null,
DEFINITIVA_MED numeric(5)    null,
COMPLEMENT_MED numeric(5)    null,
CONDICION_MED char(6)      null,
FECHAEXA_MED  datetime     null,
C_LLECTIVO_MED numeric(4)    null,
PROCESO_MED   binary(1)     null,
L1_1_MED     char(2)      null,
L1_2_MED     char(2)      null,
L1_3_MED     char(2)      null,
PL_1_MED     char(2)      null,
L2_1_MED     char(2)      null,
L2_2_MED     char(2)      null,
L2_3_MED     char(2)      null,
PL_2_MED     char(2)      null,
PF_MED       char(2)      null,
USUARIO_US_MED char(7)     null,
USUARIO_Q_MED char(7)     null,
constraint PK_MEDIO primary key (CODIGO_MED)
)
go

/*=====*/
/* Index: RELATIONSHIP_27_FK */
/*=====*/
create index RELATIONSHIP_27_FK on MEDIO (
CODIGO_APLA
)
go

/*=====*/

```

```

/* Index: RELATIONSHIP_28_FK */
/*=====*/
create index RELATIONSHIP_28_FK on MEDIO (
CODIGO_MPLA
)
go

```

```

/*=====*/
/* Index: RELATIONSHIP_29_FK */
/*=====*/
create index RELATIONSHIP_29_FK on MEDIO (
TIDO_ALU
)
go

```

```

/*=====*/
/* Table: MODALI */
/*=====*/
create table MODALI (
CODIGO_MOD char(4) not null,
CODIGO_ORIE numeric(2) null,
NOMBRE_MOD char(30) null,
NIVEL_MOD char(1) null,
ORIENTA_MOD numeric(2) null,
DURACION_MOD numeric(1) null,
COMIENZO_MOD numeric(1) null,
FINAL_MOD binary(1) null,
NOREDONDEO_MOD binary(1) null,
T_CERTI_MOD numeric(1) null,
DIGI_MAB_MOD char(2) null,
DECRI_MAB_MOD char(30) null,
CICLO_MOD char(30) null,
SI_C_MOD text null,
SI_I_MOD text null,
CERTIFI_MOD text null,
TITULO_MOD char(70) null,
USUARIO_US_MOD char(7) null,
USUARIO_Q_MOD char(7) null,
constraint PK_MODALI primary key (CODIGO_MOD)
)
go

```

```

/*=====*/
/* Index: RELATIONSHIP_24_FK */
/*=====*/
create index RELATIONSHIP_24_FK on MODALI (
CODIGO_ORIE
)
go

```

```

/*=====*/
/* Table: M_PLAN */
/*=====*/
create table M_PLAN (

```

```

CODIGO_MPLA      char(15)      not null,
CODIGO_PER       char(8)       null,
CODIGO_CPLA     char(15)      null,
MATERIA_MPLA    char(3)       null,
REFERENCIA_MPLA char(3)      null,
POSICION_MPLA   numeric(3)    null,
CARGA_MPLA      numeric(2)    null,
VANALITICO_MPLA binary(1)   null,
SEPREMEDIA_MPLA binary(1)   null,
AREACON_MPLA    char(2)       null,
NUMOLET_MPLA    char(1)       null,
PROMEXA_MPLA    char(1)       null,
constraint PK_M_PLAN primary key (CODIGO_MPLA)
)
go

```

```

/*=====*/
/* Index: RELATIONSHIP_26_FK                */
/*=====*/
create index RELATIONSHIP_26_FK on M_PLAN (
CODIGO_CPLA
)
go

```

```

/*=====*/
/* Index: RELATIONSHIP_61_FK                */
/*=====*/
create index RELATIONSHIP_61_FK on M_PLAN (
CODIGO_PER
)
go

```

```

/*=====*/
/* Table: NIVELES                          */
/*=====*/
create table NIVELES (
CODIGO_NIV      char(1)       not null,
TIDO_ALU       char(3)       null,
CODIGO_PER     char(8)       null,
NOMBRE_NIV     char(15)      null,
HS_CD_NIV      numeric(2)    null,
HS_CA_NIV      numeric(2)    null,
NUM_ESTA_NIV   char(10)      null,
CALIFICA_NIV   char(1)       null,
C_ETAPAS_NIV   numeric(1)    null,
C_MAT_REPI_NIV numeric       null,
P_REINC_NIV    numeric(2)    null,
S_REINC_NIV    numeric(2)    null,
T_REINC_NIV    numeric(2)    null,
T_SANCION_NIV  numeric(2)    null,
ORDEN_NIV     char(1)       null,
NOTAREAL_NIV   binary(1)    null,
INICIO_NIV     datetime      null,
FINAL_NIV     datetime      null,
FIN_1_NIV     datetime      null,

```

```

FIN_2_NIV      datetime      null,
FIN_3_NIV      datetime      null,
FIN_1_I_NIV    datetime      null,
FIN_2_I_NIV    datetime      null,
O_LIBRETA_NIV  char(1)        null,
UOT_MATER_NIV  char(1)        null,
LIBESPEC_NIV   binary(1)     null,
TITULO_NIV     char(63)     null,
INSPECCION_NIV char(50)     null,
USUARIO_US_NIV char(7)        null,
USUARIO_Q_NIV  char(7)        null,
constraint PK_NIVELES primary key (CODIGO_NIV)
)
go

/*=====*/
/* Index: RELATIONSHIP_11_FK                */
/*=====*/
create index RELATIONSHIP_11_FK on NIVELES (
CODIGO_PER
)
go

/*=====*/
/* Index: RELATIONSHIP_12_FK                */
/*=====*/
create index RELATIONSHIP_12_FK on NIVELES (
TIDO_ALU
)
go

/*=====*/
/* Table: NOVEDAD                          */
/*=====*/
create table NOVEDAD (
CODIGO_NOV      char(8)        not null,
CODIGO_PER      char(8)        null,
FECHA_I_NOV     datetime      null,
FECHA_F_NOV     datetime      null,
ARTICULO_NOV    char(8)        null,
CANO_NOV        char(19)       null,
CANTI_HS_M_NOV  numeric(5)     null,
CANTI_DI_M_NOV  numeric(5)     null,
HORAS_NOV       numeric(5)     null,
DIAS_NOV        numeric(5)     null,
OBSERVA_NOV     char(30)       null,
USUARIO_US_NOV  char(7)        null,
USUARIO_Q_NOV   char(7)        null,
constraint PK_NOVEDAD primary key (CODIGO_NOV)
)
go

/*=====*/
/* Index: RELATIONSHIP_53_FK                */
/*=====*/

```

```

/*=====*/
create index RELATIONSHIP_53_FK on NOVEDAD (
CODIGO_PER
)
go

```

```

/*=====*/
/* Table: OCUPA */
/*=====*/
create table OCUPA (
CODIGO_OCUP char(3) not null,
TIDO_ALU char(3) null,
DESCRI_OCUP char(20) null,
USUARIO_US_OCUP char(7) null,
USUARIO_Q_OCUP char(7) null,
constraint PK_OCUPA primary key (CODIGO_OCUP)
)
go

```

```

/*=====*/
/* Index: RELATIONSHIP_13_FK */
/*=====*/
create index RELATIONSHIP_13_FK on OCUPA (
TIDO_ALU
)
go

```

```

/*=====*/
/* Table: ORIENTA */
/*=====*/
create table ORIENTA (
CODIGO_ORIE numeric(2) not null,
TIDO_ALU char(3) null,
CODIGO_PER char(8) null,
NOMBRE_ORIE char(30) null,
SUB_NOMBRE_ORIE char(30) null,
USUARIO_US_ORIE char(7) null,
USUARIO_Q_ORIE char(7) null,
constraint PK_ORIENTA primary key (CODIGO_ORIE)
)
go

```

```

/*=====*/
/* Index: RELATIONSHIP_22_FK */
/*=====*/
create index RELATIONSHIP_22_FK on ORIENTA (
TIDO_ALU
)
go

```

```

/*=====*/
/* Index: RELATIONSHIP_23_FK */
/*=====*/

```

```

create index RELATIONSHIP_23_FK on ORIENTA (
CODIGO_PER
)
go

```

```

/*=====*/
/* Table: PERSONAL */
/*=====*/

```

```

create table PERSONAL (
TIDO_PER char(3) null,
CODIGO_PER char(8) not null,
CODIGO_TIDOC char(3) null,
CODIGO_SANG char(1) null,
CODIGO_BANC char(3) null,
CODIGO_LOC char(5) null,
CODIGO_ECIV char(1) null,
NOMBRE_PER char(30) null,
DIRECCION_PER char(30) null,
COD_CIU_PER char(5) null,
TE_PER char(15) null,
FECHANACI_PER datetime null,
SEXO_PER char(1) null,
SANGRE_PER char(1) null,
ESTA_CIVIL_PER char(1) null,
HIJOS_PER numeric(2) null,
ANTIGUEDAD_PER datetime null,
NROLEGAJO_PER char(8) null,
FECHACARPE_PER datetime null,
JUNTA_PER binary(1) null,
BORRA_PER numeric(1) null,
FECULTMOV_PER datetime null,
FOJASERVI_PER text null,
SUC_BANCO_PER char(3) null,
CUIL_PER numeric(12) null,
PASSWORD_PER char(10) null,
PREGUNTA_PER numeric(1) null,
RESPUESTA_PER char(10) null,
CREACION_PER datetime null,
EMAIL_PER char(50) null,
USUARIO_US_PER char(7) null,
USUARIO_Q_PER char(7) null,
constraint PK_PERSONAL primary key (CODIGO_PER)
)
go

```

```

/*=====*/
/* Index: RELATIONSHIP_19_FK */
/*=====*/

```

```

create index RELATIONSHIP_19_FK on PERSONAL (
CODIGO_TIDOC
)
go

```

```

/*=====*/
/* Index: RELATIONSHIP_42_FK */

```

```

/*=====*/
create index RELATIONSHIP_42_FK on PERSONAL (
CODIGO_SANG
)
go

```

```

/*=====*/
/* Index: RELATIONSHIP_43_FK */
/*=====*/
create index RELATIONSHIP_43_FK on PERSONAL (
CODIGO_LOC
)
go

```

```

/*=====*/
/* Index: RELATIONSHIP_48_FK */
/*=====*/
create index RELATIONSHIP_48_FK on PERSONAL (
CODIGO_ECIV
)
go

```

```

/*=====*/
/* Index: RELATIONSHIP_50_FK */
/*=====*/
create index RELATIONSHIP_50_FK on PERSONAL (
CODIGO_BANC
)
go

```

```

/*=====*/
/* Table: PER_ADI */
/*=====*/
create table PER_ADI (
CODIGO_PEAD char(8) not null,
CODIGO_PER char(8) null,
IDENTIFICA_PEAD char(2) null,
CODI_IDE_P_PEAD char(4) null,
CODI_IDE_S_PEAD char(4) null,
CODI_ESPE_PEAD char(10) null,
constraint PK_PER_ADI primary key (CODIGO_PEAD)
)
go

```

```

/*=====*/
/* Index: RELATIONSHIP_46_FK */
/*=====*/
create index RELATIONSHIP_46_FK on PER_ADI (
CODIGO_PER
)
go

```

```

/*=====*/
/* Table: PRECE                                     */
/*=====*/
create table PRECE (
  CODIGO_PRE      char(8)      not null,
  NOMBRE_PRE      char(30)     null,
  USUARIO_US_PRE  char(7)      null,
  USUARIO_Q_PRE   char(7)      null,
  constraint PK_PRECE primary key (CODIGO_PRE)
)
go

```

```

/*=====*/
/* Table: PRESTAMO                                 */
/*=====*/
create table PRESTAMO (
  COMPRO_PRES     numeric(8)    not null,
  TIDO_ALU        char(3)       null,
  CODIGO_PER      char(8)       null,
  FECHA_PRES      datetime      null,
  TIPO_USUA_PRES  char(1)       null,
  USUARIOS_PRES   char(8)       null,
  FECHA_VTO_PRES  datetime      null,
  ESTADO_PRES     numeric(1)    null,
  INTI_PRES       numeric(1)    null,
  USUARIO_US_PRES char(7)       null,
  USUARIO_Q_PRES  char(7)       null,
  constraint PK_PRESTAMO primary key (COMPRO_PRES)
)
go

```

```

/*=====*/
/* Index: RELATIONSHIP_33_FK                       */
/*=====*/
create index RELATIONSHIP_33_FK on PRESTAMO (
  TIDO_ALU
)
go

```

```

/*=====*/
/* Index: RELATIONSHIP_34_FK                       */
/*=====*/
create index RELATIONSHIP_34_FK on PRESTAMO (
  CODIGO_PER
)
go

```

```

/*=====*/
/* Table: REGIMEN                                 */
/*=====*/
create table REGIMEN (
  CODIGO_REGI     char(8)       not null,
  CODIGO_NOV      char(8)       null,
  NOMBRE_REGI     char(30)      null,

```

```

TIPO_REGI      char(3)      null,
INCIDENCIA_REGI  binary(1)    null,
ACUMULA_REGI    binary(1)    null,
CANTIDAD_REGI   numeric(5)   null,
PERIODO_REGI    numeric(2)   null,
COD_REG_REGI    binary(1)    null,
OBSERVA_REGI   char(20)     null,
USUARIO_US_REGI char(7)      null,
USUARIO_Q_REGI char(7)      null,
constraint PK_REGIMEN primary key (CODIGO_REGI)
)
go

/*=====*/
/* Index: RELATIONSHIP_44_FK */
/*=====*/
create index RELATIONSHIP_44_FK on REGIMEN (
CODIGO_NOV
)
go

/*=====*/
/* Table: RELATIONSHIP_3 */
/*=====*/
create table RELATIONSHIP_3 (
IDENTI_VACU char(4) not null,
CODIGO_VACALU char(8) not null,
constraint PK_RELATIONSHIP_3 primary key (IDENTI_VACU, CODIGO_VACALU)
)
go

/*=====*/
/* Index: RELATIONSHIP_3_FK */
/*=====*/
create index RELATIONSHIP_3_FK on RELATIONSHIP_3 (
IDENTI_VACU
)
go

/*=====*/
/* Index: RELATIONSHIP_4_FK */
/*=====*/
create index RELATIONSHIP_4_FK on RELATIONSHIP_3 (
CODIGO_VACALU
)
go

/*=====*/
/* Table: RELATIONSHIP_53 */
/*=====*/
create table RELATIONSHIP_53 (
CODIGO_LOC char(5) not null,
CODIGO_BANC char(3) not null,

```

```

constraint PK_RELATIONSHIP_53 primary key (CODIGO_LOC, CODIGO_BANC)
)
go

```

```

/*=====*/
/* Index: RELATIONSHIP_57_FK */
/*=====*/
create index RELATIONSHIP_57_FK on RELATIONSHIP_53 (
CODIGO_LOC
)
go

```

```

/*=====*/
/* Index: RELATIONSHIP_58_FK */
/*=====*/
create index RELATIONSHIP_58_FK on RELATIONSHIP_53 (
CODIGO_BANC
)
go

```

```

/*=====*/
/* Table: SANGRE */
/*=====*/
create table SANGRE (
CODIGO_SANG char(1) not null,
DESCRI_SANG char(5) null,
USUARIO_US_SANG char(7) null,
USUARIO_Q_SANG char(7) null,
constraint PK_SANGRE primary key (CODIGO_SANG)
)
go

```

```

/*=====*/
/* Table: SITUA */
/*=====*/
create table SITUA (
CODIGO_SITU char(1) not null,
CODIGO_PER char(8) null,
NOMBRE_SITU char(15) null,
USUARIO_US_SITU char(7) null,
USUARIO_Q_SITU char(7) null,
constraint PK_SITUA primary key (CODIGO_SITU)
)
go

```

```

/*=====*/
/* Index: RELATIONSHIP_47_FK */
/*=====*/
create index RELATIONSHIP_47_FK on SITUA (
CODIGO_PER
)
go

```

```

/*=====*/
/* Table: TIPODOCU                                     */
/*=====*/

```

```

create table TIPODOCU (
  CODIGO_TIDOC      char(3)      not null,
  TIDO_ALU          char(3)      null,
  CODIGO_PER        char(8)      null,
  NOMBRE_TIDOC      char(30)     null,
  USUARIO_US_TIDOC char(7)       null,
  USUARIO_Q_TIDOC   char(7)       null,
  constraint PK_TIPODOCU primary key (CODIGO_TIDOC)
)
go

```

```

/*=====*/
/* Index: RELATIONSHIP_18_FK                           */
/*=====*/

```

```

create index RELATIONSHIP_18_FK on TIPODOCU (
  TIDO_ALU
)
go

```

```

/*=====*/
/* Index: RELATIONSHIP_20_FK                           */
/*=====*/

```

```

create index RELATIONSHIP_20_FK on TIPODOCU (
  CODIGO_PER
)
go

```

```

/*=====*/
/* Table: TITULOS                                       */
/*=====*/

```

```

create table TITULOS (
  INVENTARIO_TIT      numeric(5)      not null,
  CODIGO_AUT           numeric(4)      null,
  CODIGO_EDIT          numeric(4)      null,
  COMPRO_CUPRE         numeric(8)      null,
  MATERIAL_TIT         numeric(2)      null,
  CLAVE_TIT            char(17)        null,
  ISBN_TIT             char(13)        null,
  CODIGOBARRA_TIT      char(13)        null,
  AUTOR_TIT            numeric(4)      null,
  TITULO_TIT           char(40)        null,
  SUB_TITULO_TIT       char(30)        null,
  EDITORIAL_TIT        numeric(4)      null,
  EDICION_TIT          numeric(2)      null,
  PAGINAS_TIT          numeric(4)      null,
  LUGAR_EDI_TIT        char(15)        null,
  ANO_EDI_TIT          numeric(4)      null,
  FORMA INGR_TIT      char(8)         null,
  PRECIO_UL_TIT        numeric(10)     null,
  FECHA_ING_TIT        datetime        null,
  UBICACION_TIT        char(20)        null,

```

```

OBSERVA_TIT      char(30)      null,
INDICI_TIT       text          null,
FECULTMOV_TIT    datetime      null,
BORRA_TIT        numeric(1)     null,
USUARIO_US_TIT   char(7)        null,
USUARIO_Q_TIT    char(7)        null,
constraint PK_TITULOS primary key (INVENTARIO_TIT)
)
go

```

```

/*=====*/
/* Index: RELATIONSHIP_36_FK                */
/*=====*/
create index RELATIONSHIP_36_FK on TITULOS (
COMPRO_CUPRE
)
go

```

```

/*=====*/
/* Index: RELATIONSHIP_37_FK                */
/*=====*/
create index RELATIONSHIP_37_FK on TITULOS (
CODIGO_EDIT
)
go

```

```

/*=====*/
/* Index: RELATIONSHIP_38_FK                */
/*=====*/
create index RELATIONSHIP_38_FK on TITULOS (
CODIGO_AUT
)
go

```

```

/*=====*/
/* Table: TITULO_PROF                       */
/*=====*/
create table TITULO_PROF (
CODIGO_TIPR      char(4)        not null,
CODIGO_PEAD      char(8)        null,
NOMBRE_TIPR      char(30)       null,
USUARIO_US_TIPR  char(7)        null,
USUARIO_Q_TIPR   char(7)        null,
constraint PK_TITULO_PROF primary key (CODIGO_TIPR)
)
go

```

```

/*=====*/
/* Index: RELATIONSHIP_45_FK                */
/*=====*/
create index RELATIONSHIP_45_FK on TITULO_PROF (
CODIGO_PEAD
)

```

go

```
/*=====*/
/* Table: TURNOS */
/*=====*/
create table TURNOS (
  CODIGO_TURN char(1) not null,
  TIDO_ALU char(3) null,
  CODIGO_PER char(8) null,
  NOMBRE_TURN char(15) null,
  DIGITO_TURN char(2) null,
  N_MATRICU_TURN numeric(6) null,
  USUARIO_US_TURN char(7) null,
  USUARIO_Q_TURN char(7) null,
  constraint PK_TURNOS primary key (CODIGO_TURN)
)
go
```

```
/*=====*/
/* Index: RELATIONSHIP_15_FK */
/*=====*/
create index RELATIONSHIP_15_FK on TURNOS (
  TIDO_ALU
)
go
```

```
/*=====*/
/* Index: RELATIONSHIP_16_FK */
/*=====*/
create index RELATIONSHIP_16_FK on TURNOS (
  CODIGO_PER
)
go
```

```
/*=====*/
/* Table: T_FALTAS */
/*=====*/
create table T_FALTAS (
  CODIGO_TFAL numeric(2) not null,
  CODIGO_AUSE char(8) null,
  NOMBRE_TFAL char(30) null,
  CANTIDAD_TFAL numeric(5) null,
  ACUMULA_TFAL binary(1) null,
  HASTACU_FALT numeric(2) null,
  INCIDE_TFAL binary(1) null,
  TIPOFALTA_TFAL numeric(1) null,
  TARDJUSTI_TFAL binary(1) null,
  TARDACUMU_TFAL numeric(4) null,
  USUARIO_US_TFAL char(7) null,
  USUARIO_Q_TFAL char(7) null,
  constraint PK_T_FALTAS primary key (CODIGO_TFAL)
)
go
```

```

/*=====*/
/* Index: RELATIONSHIP_6_FK */
/*=====*/
create index RELATIONSHIP_6_FK on T_FALTAS (
CODIGO_AUSE
)
go

```

```

/*=====*/
/* Table: VACALU */
/*=====*/
create table VACALU (
CODIGO_VACALU char(8) not null,
TIDO_ALU char(3) null,
IDENTI_VACALU char(4) null,
constraint PK_VACALU primary key (CODIGO_VACALU)
)
go

```

```

/*=====*/
/* Index: RELATIONSHIP_5_FK */
/*=====*/
create index RELATIONSHIP_5_FK on VACALU (
TIDO_ALU
)
go

```

```

/*=====*/
/* Table: VACUNA */
/*=====*/
create table VACUNA (
IDENTI_VACU char(4) not null,
DESCRI_VACU char(20) null,
USUARIO_US_VACU char(7) null,
USUARIO_Q_VACU char(7) null,
constraint PK_VACUNA primary key (IDENTI_VACU)
)
go

```

```

alter table ALUMNOS
add constraint FK_ALUMNOS_RELATIONS_TURNOS foreign key (CODIGO_TURN)
references TURNOS (CODIGO_TURN)
go

```

```

alter table ALUMNOS
add constraint FK_ALUMNOS_RELATIONS_TIPODOCU foreign key (CODIGO_TIDOC)
references TIPODOCU (CODIGO_TIDOC)
go

```

```

alter table ALUMNOS
add constraint FK_ALUMNOS_RELATIONS_ORIENTA foreign key (CODIGO_ORIE)

```

```

    references ORIENTA (CODIGO_ORIE)
go

alter table ALUMNOS
    add constraint FK_ALUMNOS_RELATIONS_CURSOS foreign key (CODIGO_CURS)
    references CURSOS (CODIGO_CURS)
go

alter table ALUMNOS
    add constraint FK_ALUMNOS_RELATIONS_SANGRE foreign key (CODIGO_SANG)
    references SANGRE (CODIGO_SANG)
go

alter table ALUMNOS
    add constraint FK_ALUMNOS_RELATIONS_LOCALI foreign key (CODIGO_LOC)
    references LOCALI (CODIGO_LOC)
go

alter table AUSENTES
    add constraint FK_AUSENTES_RELATIONS_ALUMNOS foreign key (TIDO_ALU)
    references ALUMNOS (TIDO_ALU)
go

alter table A_PLAN
    add constraint FK_A_PLAN_RELATIONS_C_PLAN foreign key (CODIGO_CPLA)
    references C_PLAN (CODIGO_CPLA)
go

alter table A_PLAN
    add constraint FK_A_PLAN_RELATIONS_PERSONAL foreign key (CODIGO_PER)
    references PERSONAL (CODIGO_PER)
go

alter table BANCOS
    add constraint FK_BANCOS_RELATIONS_PERSONAL foreign key (CODIGO_PER)
    references PERSONAL (CODIGO_PER)
go

alter table CARGOS
    add constraint FK_CARGOS_RELATIONS_PERSONAL foreign key (CODIGO_PER)
    references PERSONAL (CODIGO_PER)
go

alter table CONDICIO
    add constraint FK_CONDICIO_RELATIONS_PERSONAL foreign key (CODIGO_PER)
    references PERSONAL (CODIGO_PER)
go

```

```
alter table CUERPRES
  add constraint FK_CUERPRES_RELATIONS_PRESTAMO foreign key (COMPRO_PRES)
  references PRESTAMO (COMPRO_PRES)
go
```

```
alter table CURSOS
  add constraint FK_CURSOS_RELATIONS_ALUMNOS foreign key (TIDO_ALU)
  references ALUMNOS (TIDO_ALU)
go
```

```
alter table CURSOS
  add constraint FK_CURSOS_RELATIONS_C_PRECE foreign key (CODIGO_CPRE)
  references C_PRECE (CODIGO_CPRE)
go
```

```
alter table CURSOS
  add constraint FK_CURSOS_RELATIONS_PERSONAL foreign key (CODIGO_PER)
  references PERSONAL (CODIGO_PER)
go
```

```
alter table C_PLAN
  add constraint FK_C_PLAN_RELATIONS_MODALI foreign key (CODIGO_MOD)
  references MODALI (CODIGO_MOD)
go
```

```
alter table C_PRECE
  add constraint FK_C_PRECE_RELATIONS_PRECE foreign key (CODIGO_PRE)
  references PRECE (CODIGO_PRE)
go
```

```
alter table E_CIVIL
  add constraint FK_E_CIVIL_RELATIONS_PERSONAL foreign key (CODIGO_PER)
  references PERSONAL (CODIGO_PER)
go
```

```
alter table HORAS
  add constraint FK_HORAS_RELATIONS_PERSONAL foreign key (CODIGO_PER)
  references PERSONAL (CODIGO_PER)
go
```

```
alter table HS_PROFE
  add constraint FK_HS_PROFE_RELATIONS_PERSONAL foreign key (CODIGO_PER)
  references PERSONAL (CODIGO_PER)
go
```

```
alter table MATERIAL
  add constraint FK_MATERIAL_RELATIONS_TITULOS foreign key (INVENTARIO_TIT)
  references TITULOS (INVENTARIO_TIT)
go
```

```

alter table MEDIO
  add constraint FK_MEDIO_RELATIONS_A_PLAN foreign key (CODIGO_APLA)
    references A_PLAN (CODIGO_APLA)
go

alter table MEDIO
  add constraint FK_MEDIO_RELATIONS_M_PLAN foreign key (CODIGO_MPLA)
    references M_PLAN (CODIGO_MPLA)
go

alter table MEDIO
  add constraint FK_MEDIO_RELATIONS_ALUMNOS foreign key (TIDO_ALU)
    references ALUMNOS (TIDO_ALU)
go

alter table MODALI
  add constraint FK_MODALI_RELATIONS_ORIENTA foreign key (CODIGO_ORIE)
    references ORIENTA (CODIGO_ORIE)
go

alter table M_PLAN
  add constraint FK_M_PLAN_RELATIONS_C_PLAN foreign key (CODIGO_CPLA)
    references C_PLAN (CODIGO_CPLA)
go

alter table M_PLAN
  add constraint FK_M_PLAN_RELATIONS_PERSONAL foreign key (CODIGO_PER)
    references PERSONAL (CODIGO_PER)
go

alter table NIVELES
  add constraint FK_NIVELES_RELATIONS_PERSONAL foreign key (CODIGO_PER)
    references PERSONAL (CODIGO_PER)
go

alter table NIVELES
  add constraint FK_NIVELES_RELATIONS_ALUMNOS foreign key (TIDO_ALU)
    references ALUMNOS (TIDO_ALU)
go

alter table NOVEDAD
  add constraint FK_NOVEDAD_RELATIONS_PERSONAL foreign key (CODIGO_PER)
    references PERSONAL (CODIGO_PER)
go

alter table OCUPA
  add constraint FK_OCUPA_RELATIONS_ALUMNOS foreign key (TIDO_ALU)

```

```
references ALUMNOS (TIDO_ALU)
go
```

```
alter table ORIENTA
add constraint FK_ORIENTA_RELATIONS_ALUMNOS foreign key (TIDO_ALU)
references ALUMNOS (TIDO_ALU)
go
```

```
alter table ORIENTA
add constraint FK_ORIENTA_RELATIONS_PERSONAL foreign key (CODIGO_PER)
references PERSONAL (CODIGO_PER)
go
```

```
alter table PERSONAL
add constraint FK_PERSONAL_RELATIONS_TIPODOCU foreign key (CODIGO_TIDOC)
references TIPODOCU (CODIGO_TIDOC)
go
```

```
alter table PERSONAL
add constraint FK_PERSONAL_RELATIONS_SANGRE foreign key (CODIGO_SANG)
references SANGRE (CODIGO_SANG)
go
```

```
alter table PERSONAL
add constraint FK_PERSONAL_RELATIONS_LOCALI foreign key (CODIGO_LOC)
references LOCALI (CODIGO_LOC)
go
```

```
alter table PERSONAL
add constraint FK_PERSONAL_RELATIONS_E_CIVIL foreign key (CODIGO_ECIV)
references E_CIVIL (CODIGO_ECIV)
go
```

```
alter table PERSONAL
add constraint FK_PERSONAL_RELATIONS_BANCOS foreign key (CODIGO_BANC)
references BANCOS (CODIGO_BANC)
go
```

```
alter table PER_ADI
add constraint FK_PER_ADI_RELATIONS_PERSONAL foreign key (CODIGO_PER)
references PERSONAL (CODIGO_PER)
go
```

```
alter table PRESTAMO
add constraint FK_PRESTAMO_RELATIONS_ALUMNOS foreign key (TIDO_ALU)
references ALUMNOS (TIDO_ALU)
go
```

```

alter table PRESTAMO
  add constraint FK_PRESTAMO_RELATIONS_PERSONAL foreign key (CODIGO_PER)
    references PERSONAL (CODIGO_PER)
go

alter table REGIMEN
  add constraint FK_REGIMEN_RELATIONS_NOVEDAD foreign key (CODIGO_NOV)
    references NOVEDAD (CODIGO_NOV)
go

alter table RELATIONSHIP_3
  add constraint FK_RELATION_RELATIONS_VACUNA foreign key (IDENTI_VACU)
    references VACUNA (IDENTI_VACU)
go

alter table RELATIONSHIP_3
  add constraint FK_RELATION_RELATIONS_VACALU foreign key (CODIGO_VACALU)
    references VACALU (CODIGO_VACALU)
go

alter table RELATIONSHIP_53
  add constraint FK_RELATION_RELATIONS_LOCALI foreign key (CODIGO_LOC)
    references LOCALI (CODIGO_LOC)
go

alter table RELATIONSHIP_53
  add constraint FK_RELATION_RELATIONS_BANCOS foreign key (CODIGO_BANC)
    references BANCOS (CODIGO_BANC)
go

alter table SITUA
  add constraint FK_SITUA_RELATIONS_PERSONAL foreign key (CODIGO_PER)
    references PERSONAL (CODIGO_PER)
go

alter table TIPODOCU
  add constraint FK_TIPODOCU_RELATIONS_ALUMNOS foreign key (TIDO_ALU)
    references ALUMNOS (TIDO_ALU)
go

alter table TIPODOCU
  add constraint FK_TIPODOCU_RELATIONS_PERSONAL foreign key (CODIGO_PER)
    references PERSONAL (CODIGO_PER)
go

alter table TITULOS
  add constraint FK_TITULOS_RELATIONS_CUERPRES foreign key (COMPRO_CUPRE)
    references CUERPRES (COMPRO_CUPRE)
go

```

```
alter table TITULOS
  add constraint FK_TITULOS_RELATIONS_EDITORIA foreign key (CODIGO_EDIT)
  references EDITORIA (CODIGO_EDIT)
go
```

```
alter table TITULOS
  add constraint FK_TITULOS_RELATIONS_AUTORES foreign key (CODIGO_AUT)
  references AUTORES (CODIGO_AUT)
go
```

```
alter table TITULO_PROF
  add constraint FK_TITULO_P_RELATIONS_PER_ADI foreign key (CODIGO_PEAD)
  references PER_ADI (CODIGO_PEAD)
go
```

```
alter table TURNOS
  add constraint FK_TURNOS_RELATIONS_ALUMNOS foreign key (TIDO_ALU)
  references ALUMNOS (TIDO_ALU)
go
```

```
alter table TURNOS
  add constraint FK_TURNOS_RELATIONS_PERSONAL foreign key (CODIGO_PER)
  references PERSONAL (CODIGO_PER)
go
```

```
alter table T_FALTAS
  add constraint FK_T_FALTAS_RELATIONS_AUSENTES foreign key (CODIGO_AUSE)
  references AUSENTES (CODIGO_AUSE)
go
```

```
alter table VACALU
  add constraint FK_VACALU_RELATIONS_ALUMNOS foreign key (TIDO_ALU)
  references ALUMNOS (TIDO_ALU)
go
```

DOCUMENTACION ANALIZADA