



**UNIVERSIDAD NACIONAL
AUTÓNOMA DE MÉXICO**

FACULTAD DE INGENIERÍA

**SISTEMA DE INFORMACIÓN DEL
PROGRAMA DE ESTÍMULOS Y
RECONOCIMIENTO AL PERSONAL
ACADÉMICO EMÉRITO (PERPAE)**

TESIS

**QUE PARA OBTENER EL TÍTULO DE:
INGENIERO EN COMPUTACIÓN**

**PRESENTA:
SANDRA LILIANA RAMÍREZ MORA**

**DIRECTOR DE TESIS:
DR. VÍCTOR HUGO JACOBO ARMENDÁRIZ**



CIUDAD UNIVERSITARIA, OCTUBRE 2012

ÍNDICE

PRÓLOGO	4
CAPÍTULO I. ANTECEDENTES.....	5
1.1 Dirección General de Asuntos del personal académico (DGAPA).....	6
1.2 Programa de Estímulos y Reconocimiento al Personal Académico Emérito (PERPAE)	8
1.3 PROPUESTA DE TESIS.....	9
CAPÍTULO II. MARCO TEÓRICO	11
2.1 SISTEMAS DE INFORMACIÓN.....	12
2.2 CICLO DE VIDA DE DESARROLLO DE SOFTWARE	14
2.3 METODOLOGÍA DE DESARROLLO DE SOFTWARE.....	16
2.4 LENGUAJES DE PROGRAMACIÓN	21
2.5 LENGUAJE UNIFICADO DE MODELADO	23
2.6 BASES DE DATOS.....	28
CAPÍTULO III. PREPARACIÓN.....	30
3.1 DEFINICIÓN DEL SISTEMA.....	31
3.2 <i>PRODUCT BACKLOG</i> DEL SISTEMA	33
3.3 DEFINICIÓN DE ACTIVIDADES y PLAN DE ENTREGA	38
CAPÍTULO VI. ANÁLISIS Y DISEÑO	39
4.1 ANÁLISIS DE REQUERIMIENTOS	40
4.2 MODELO DE DATOS	53
4.3 ELABORACIÓN DE CASOS DE USO	57
4.4 DISEÑO DE LA INTERFAZ GRÁFICA.....	63
4.5 DISEÑO DE REPORTE.....	67
4.6 DISEÑO DE GRÁFICAS.....	68
CAPÍTULO V. DESARROLLO.....	69
5.1 TECNOLOGÍA Y LOGÍSTICA	70
5.2 IMPLEMENTACIÓN DE LA BASE DE DATOS.....	74
5.3 GENERACIÓN Y CONFIGURACIÓN DEL PROYECTO	77
5.4 MÉTODO DE PROGRAMACIÓN.....	82
5.5 CODIFICACIÓN	84
5.6 DESARROLLO DE REPORTE Y GRÁFICAS	93
CAPÍTULO VI. PRUEBAS E IMPLANTACIÓN	96
6.1 PRUEBAS	97
6.2 IMPLANTACIÓN	100
CONCLUSIONES	101
ANEXOS.....	103

DICCIONARIO DE DATOS.....	103
INTERFACES GRÁFICAS	118
MODELO DE DATOS	118
GLOSARIO	144
BIBLIOGRAFÍA Y REFERENCIAS ELECTRÓNICAS	146

PRÓLOGO

El objetivo de este trabajo de tesis es describir el desarrollo de un sistema para la gestión de información del Programa de Estímulos y Reconocimiento al Personal Académico Emérito (PERPAE), el cual es gestionado por la Dirección General de Asuntos del Personal Académico (DGAPA).

En este trabajo se presentan las actividades de cada una de las fases de desarrollo del sistema mencionado, en el marco de la ingeniería en computación y en específico de la ingeniería de software.

La estructura de este trabajo cuenta con seis capítulos que abordan distintos aspectos sobre el desarrollo y marco de desarrollo del sistema de información.

En el primer capítulo llamado “Antecedentes”, se describen algunos aspectos de la Dirección General de Asuntos del Personal Académico (DGAPA) y del Programa de Estímulos y Reconocimiento al Personal Académico Emérito (PERPAE). Además, se describen aspectos importantes en relación al presente trabajo de tesis.

En el segundo capítulo llamado “Marco teórico”, se presentan de manera general conceptos teóricos referentes a la Ingeniería en Computación que se emplean durante las fases de desarrollo del sistema de información.

El tercer capítulo llamado “Preparación”, tiene como objetivo describir las actividades realizadas al inicio del ciclo de vida de desarrollo del sistema de información, las cuales sirvieron como base para realizar actividades de análisis y diseño, y permitieron gestionar aspectos importantes del sistema.

En el cuarto capítulo llamado “Análisis y diseño”, se presentan las actividades realizadas para detallar los requerimientos del sistema de información y modelarlo de una forma adecuada.

El quinto capítulo llamado “Desarrollo”, tiene como fin presentar el proceso de construcción del sistema de información, en el que se realizaron tareas de codificación y se generaron los componentes del sistema.

En el sexto y último capítulo llamado “Pruebas e implantación”, se describen las pruebas realizadas al sistema de información, además de presentar su implantación en el hardware adecuado.

CAPÍTULO I

ANTECEDENTES

Con el objetivo de introducir al lector en el contexto en que se desarrolló este trabajo, se presenta información relacionada con la Dirección General de Asuntos del Personal Académico (DGAPA), encargada de gestionar el Programa de Estímulos y Reconocimiento al Personal Académico Emérito (PERPE), descrito también en este capítulo.

Por otra parte, se presentan aspectos que anteceden a la realización de esta tesis, como el problema que se buscó solucionar, la hipótesis planteada y la justificación para realizar este trabajo.

1.1 Dirección General de Asuntos del personal académico (DGAPA)

La Dirección General de Asuntos del Personal Académico (DGAPA) es una dependencia de la UNAM, creada en 1977 con el propósito de integrar las actividades de apoyo al personal académico.

1.1.1 Objetivos generales

Los objetivos generales de la DGAPA son:

- Contribuir al incremento de la productividad y calidad de las labores docentes y de investigación, a través de programas de fortalecimiento, estabilidad, reconocimiento y estímulo del personal académico de la UNAM;
- Elevar el nivel académico del personal académico de la UNAM mediante el otorgamiento de becas y apoyos para la realización de estudios de posgrado y estancias de investigación, e incidir en el reforzamiento de la planta académica de la UNAM gracias a la incorporación de recursos humanos altamente calificados;
- Impulsar el desarrollo de proyectos de investigación básica, aplicada y multidisciplinaria en todas las áreas del conocimiento, consolidando a los académicos de más larga trayectoria y creando nuevos cuadros de investigadores; asimismo, actualizar la labor docente de los profesores universitarios en los niveles educativos de bachillerato y licenciatura;
- Elaborar los estudios pertinentes para determinar la incidencia de los programas académicos de la dependencia en el establecimiento de redes de investigación, de docencia y de becarios, en una dimensión nacional e internacional;
- Generar la información relevante sobre el personal académico que permita la planeación y la toma de decisiones por parte de las instancias universitarias a nivel central y local.

1.1.2 Funciones

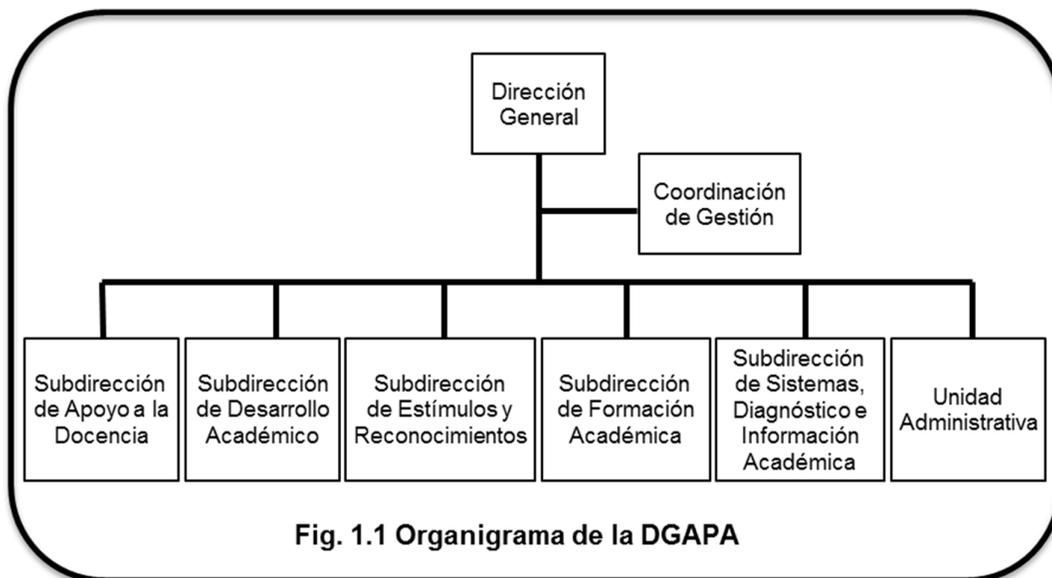
La DGAPA se ocupa de los siguientes asuntos que afectan la vida académica de distintas entidades de la UNAM:

- Ingreso, promoción y permanencia del personal académico, a través de la observancia de los procedimientos estatutarios que establece la Universidad; y mediante la integración de Comisiones Dictaminadoras designadas por cuerpos colegiados;

- Desarrollo y formación del personal académico de la Universidad, a través de programas de becas de posgrado, nacionales e internacionales, de estancias de investigación en el extranjero y posdoctorados;
- Reconocimiento y estímulo para el personal académico del más alto nivel y trayectoria, así como para los jóvenes académicos, mediante la promoción y administración de distintos programas que inciden en las percepciones académicas del personal o que premian su calidad y esfuerzo académicos;
- Actualización y perfeccionamiento de los conocimientos de los profesores, para elevar la calidad de la educación que se imparte en las diversas entidades académicas a través de sus planes de estudio correspondientes;
- Impulso a la investigación colectiva, multidisciplinaria e interinstitucional que se realiza en la Universidad en las distintas áreas del conocimiento, mediante la asignación de recursos financieros que derivan de la evaluación por pares dentro de órganos colegiados;
- Elaboración de bases de datos y difusión sistemática de la información sobre el personal académico para que las distintas dependencias universitarias puedan llevar a cabo la planeación de sus funciones específicas y la toma de decisiones.

1.1.3 Organigrama de la DGAPA

La DGAPA esta coordinada por una dirección general y una dirección de gestión, y dividida en subdirecciones y una unidad administrativa, encargadas de funciones específicas relacionadas con el personal académico de la UNAM (Fig. 1.1).



1.2 Programa de Estímulos y Reconocimiento al Personal Académico Emérito (PERPAE)

1.2.1 Antecedentes

Desde 1941, la Universidad Nacional Autónoma de México otorga la categoría de profesor o investigador emérito, a destacados miembros de su personal académico que han realizado una obra de valía excepcional y han prestado servicio a la institución con gran dedicación durante, cuando menos, treinta años. La UNAM, comprometida en apoyar al personal emérito, emitió el Acuerdo de Estímulos y Reconocimiento al Personal Académico Emérito, fundamento del Programa de Estímulos y Reconocimiento al Personal Académico Emérito (PERPAE).

1.2.2 Objetivo

Otorgar al personal académico emérito un apoyo económico, un seguro de gastos médicos, así como la difusión de su obra y pensamiento.

1.2.3 Descripción

El PERPAE es un programa de carácter permanente que va dirigido a quienes han sido designados profesores o investigadores eméritos por el Consejo Universitario.

Entre las acciones contempladas en el programa y como parte del reconocimiento, se otorga un apoyo económico y un seguro de gastos médicos mayores; así mismo, el programa considera la difusión de la obra y pensamiento de tan distinguidos universitarios.

El apoyo económico se determina de acuerdo al nombramiento y a la situación laboral en que se encuentre el emérito. Se otorga de la siguiente manera:

- Personal emérito activo de tiempo completo: En los casos que el emérito tenga el nombramiento de titular de tiempo completo, tiene el derecho de recibir un estímulo equivalente al nivel II del Programa Reconocimiento Catedrático UNAM, más la repercusión en su antigüedad académica, integrado a su sueldo y pagado quincenalmente.
- Personal emérito activo y jubilado activo: Los eméritos activos y jubilados activos con nombramiento de medio tiempo y horas asignatura, les corresponde un bono anual equivalente a 5 salarios del titular "C" de tiempo completo y se paga en dos exhibiciones semestrales (mayo y noviembre).
- Personal emérito jubilado: En el caso del personal cuya situación laboral sea jubilado, se les otorgará una gratificación mensual en forma vitalicia.

1.3 PROPUESTA DE TESIS

A continuación se presentan algunos conceptos que fueron planteados antes de proceder con la elaboración de este trabajo de tesis.

1.3.1 Problemática a solucionar

La Dirección General de Asuntos del Personal Académico (DGAPA) está encargada de la administración de diversos programas relacionados con el personal académico de la UNAM, en donde el manejo de información es una cuestión importante.

El Programa de Estímulos y Reconocimiento al Personal Académico Emérito (PERPAE), administrado por la DGAPA, emplea información que requiere ser gestionada de manera sencilla, eficaz y confiable por medio de una herramienta tecnológica.

La DGAPA ha empleado en diversas ocasiones herramientas de apoyo para la administración de información del PERPAE, pero con el paso del tiempo, se han tornado tecnológicamente obsoletas y poco prácticas, además de no cubrir con todas las características funcionales requeridas.

Debido a las necesidades de la DGAPA, se requiere contar con una herramienta integral, que además de apoyar en la administración de información del PERPAE, sea adaptable a las tecnologías y estándares utilizados en la dependencia.

1.3.2 Objetivo del presente trabajo de tesis

Desarrollar un sistema informático funcional que apoye en la administración de información del PERPAE de una manera segura, sencilla y eficaz, y con las características necesarias para su adaptación tecnológica a la DGAPA.

1.3.3 Hipótesis

Es posible el desarrollo de un sistema de información que apoye en los procesos relacionados con la administración de información del PERPAE de una manera útil, funcional y sencilla. Por otra parte, es viable la adaptación del sistema mencionado a las tecnologías utilizadas en la DGAPA, lo que facilite su gestión y mantenimiento.

1.3.4 Justificación

Existen muchos sistemas de información desarrollados en la DGAPA, encargados de brindar apoyo en la administración de información de una manera eficaz. Estos sistemas operan con éxito y son herramientas importantes para la realización de procesos no solo en la DGAPA, si no en la UNAM en general.

Por otra parte, el haber finalizado la carrera de Ingeniería en Computación, el estudio de diversos temas referentes a la Ingeniería de Software, y el haber desarrollado anteriormente herramientas de software, me da el sustento teórico y práctico para el desarrollo del sistema de información planteado.

Además de lo anterior, es importante mencionar que la DGAPA me ha facilitado la infraestructura y apoyo necesario para el desarrollo del sistema en cuestión.

1.3.5 Campo de acción

Tomando en cuenta el área de la Ingeniería, la presente tesis se centra en el área de la Ingeniería en Computación, y en específico, en la Ingeniería de Software. Algunos temas de informática en los que se centra el desarrollo del sistema son: bases de datos, programación y ciclo de vida de software.

CAPÍTULO II

MARCO TEÓRICO

Con la finalidad de introducir al lector en el marco teórico correspondiente al desarrollo de software, y en específico, al desarrollo del sistema de información en cuestión, se presentan diversos conceptos relacionados con la Ingeniería en Computación.

Algunos de los temas descritos en este capítulo son: sistemas de información, ciclo de vida de desarrollo de software, lenguajes de programación y bases de datos.

2.1 SISTEMAS DE INFORMACIÓN

En informática, un sistema de información es un conjunto de elementos que operan de forma organizada y cuyos objetivos son:

- Apoyar en las estrategias de una organización.
- Utilizar la información como recurso fundamental para el apoyo en la toma de decisiones.
- Automatizar actividades y procesos.
- Apoyar en la administración de información de manera adecuada y oportuna.

2.1.1 Características

Los sistemas de información pueden representar herramientas de gran utilidad para las organizaciones, por lo cual, es indispensable que cuenten con características que los hagan herramientas de calidad. Estas características son:

- Ser funcionales, útiles, confiables, exactos y amigables.
- Contar con un adecuado rendimiento.
- Proporcionar seguridad y disponibilidad a la información.
- Adaptarse a la evolución de la organización.

2.1.2 Funciones

Los sistemas de información realizan distintas actividades que resultan indispensables para lograr la adecuada gestión de información. Algunas de estas actividades son la captura, registro, edición, actualización, clasificación, ordenamiento, almacenamiento, procesamiento, transmisión y presentación de información.

2.1.3 Componentes

Existen distintos componentes que integran un sistema de información y que son necesarios para el funcionamiento de los mismos. Cada uno de dichos componentes realiza un papel fundamental en la operación del sistema y son:

- Herramientas tecnológicas: Son el conjunto de componentes de hardware y software que dan soporte al sistema de información.

- Sistema de datos: Información, la forma y el medio en donde es almacenada.
- Personales: Son fuentes de entrada y receptores de salidas. Realizan los procesos relacionados con la información. Son también aquellos que desarrollan y dan mantenimiento y soporte al sistema.
- Procesos: Elementos y actividades que obtienen y generan información.
- Procedimientos administrativos: Reglas y políticas que rigen el comportamiento de los usuarios frente al sistema.

2.1.4 Ventajas

Existen muchas razones por las que las organizaciones utilizan sistemas de información para apoyo en sus actividades, y se debe en gran parte a las ventajas que ofrecen. Algunas de estas ventajas son:

- Apoyan en la toma de decisiones y en el control de las actividades de una organización.
- Automatizan y aumentan la efectividad en los procesos.
- Pueden integrar diferentes áreas de una organización y tecnologías de vanguardia en su implementación.
- Proveen ventajas competitivas y valor agregado.
- Pueden ofrecer disponibilidad de la información en tiempo real.
- Permiten disminuir tiempo, costos, errores y recursos innecesarios en la realización de actividades.
- Solucionan problemas de comunicación y facilitan el intercambio y acceso a la información.

2.1.5 Desventajas

Existen algunas desventajas para el uso de sistemas de información, que aunque pueden ser controladas, se deben tomar en cuenta. Estas desventajas son:

- Se puede emplear una gran cantidad de tiempo y costo para su implementación.
- Se puede generar una resistencia al cambio por parte de los usuarios.
- Se pueden generar problemas de hardware, software o seguridad en la información.

2.2 CICLO DE VIDA DE DESARROLLO DE SOFTWARE

El desarrollo de todo software, incluyendo sistemas de información, debe pasar por una serie de fases, desde la idea o necesidad inicial, hasta su finalización y mantenimiento. Estas fases están dirigidas a garantizar el buen desarrollo del software, ya que en cada una de ellas, se realizan actividades que están enfocadas a cubrir distintas particularidades del software.

Un modelo de ciclo de vida describe y define el orden de ejecución de las fases para el desarrollo de software. También apoya en la administración y provee un marco de trabajo de acuerdo a una metodología detallada.

Existen diferentes modelos de ciclo de vida, caracterizados por el orden en que se llevan a cabo las etapas de desarrollo. Algunos de los ciclos de vida más comunes son: cascada, espiral, incremental, iterativo y por prototipos.

2.2.1 Fases del ciclo de vida de software

A continuación se describen las fases más comunes que emplean los modelos de ciclo de vida de software.

Análisis

Es la fase inicial del desarrollo de software en la que se realizan distintas actividades para determinar las características del sistema a desarrollar.

La ingeniería de requerimientos es la base del análisis, ya que permite obtener la información necesaria para entender lo que requiere el cliente y permite estudiar la problemática a resolver, tomando en cuenta el entorno en el que se encuentra.

El proceso de ingeniería de requerimientos incluye la identificación, análisis, especificación, validación y gestión de requerimientos.

Diseño

Es la etapa en la que se determina como se va a desarrollar el software partiendo del análisis realizado.

Puede dividirse en diferentes actividades, tales como: diseño de datos, diseño arquitectónico, diseño de la interfaz gráfica y diseño a nivel de componentes.

En esta etapa es importante determinar las herramienta de hardware y software que se van a emplear para el desarrollo del software, procurando que sean las adecuadas para la construcción del mismo.

Codificación o programación

Es la etapa de construcción del software, en la cual se desarrollan las interfaces gráficas, se genera el código y se implementan los módulos necesarios para el funcionamiento del software.

Es importante determinar un lenguaje de programación adecuado y herramientas tecnológicas que permitan el correcto desarrollo del software.

Distribución

Es la distribución e instalación del sistema en el hardware adecuado, y la integración del mismo con los sistemas cooperativos.

Pruebas

Es la etapa en donde se realizan actividades para garantizar el correcto funcionamiento del sistema bajo distintas circunstancias. Existen distintos tipos de pruebas y depende de las características del sistema para seleccionar las más indicadas.

Las pruebas pueden realizarse en cada una de las etapas del desarrollo o al finalizar la implementación o distribución. Esto depende de las prácticas y metodologías a las que se apegan el desarrollo del software.

Mantenimiento

Es la actualización, verificación y validación constante del sistema una vez que ha sido puesto en marcha. Surge debido a la aparición de nuevos requerimientos o porque han sido encontradas fallas en el sistema. En esta etapa es importante contar con la documentación adecuada para garantizar un buen mantenimiento del sistema.

2.3 METODOLOGÍA DE DESARROLLO DE SOFTWARE

Una metodología de desarrollo de software es un conjunto de procedimientos, técnicas, herramientas y documentos que sirven como base para el desarrollo de un software. Una metodología puede seguir uno o varios modelos de ciclo de vida e indica como deben realizarse las tareas en el mismo. Los objetivos de una metodología de desarrollo de software son:

- Definir las etapas en las que se debe dividir un proyecto, las tareas y actividades que se realizan en cada etapa del proyecto, las entradas y salidas propias de cada fase, los criterios para pasar de una fase a otra y las restricciones que se deben aplicar.
- Proporcionar herramientas para el control, revisión, planificación, organización y mantenimiento del software.
- Apoyar en el desarrollo de software de calidad, en un tiempo apropiado y con costos aceptables.
- Proveer durante el desarrollo del software, la documentación adecuada para facilitar su mantenimiento.
- Identificar de forma oportuna, los cambios que puedan presentarse en el software.
- Satisfacer a todo el personal involucrado con el software.

2.3.1 Tipos de metodologías

Existen diversas metodologías para el desarrollo de un sistema de software, entre las cuales se encuentran las metodologías tradicionales y las metodologías ágiles.

Las metodologías tradicionales se caracterizan por su planeación y análisis exhaustivo con el objetivo de conseguir un producto eficiente a través de la asignación de roles, actividades, procesos, documentación, artefactos y herramientas.

Las metodologías ágiles promueven la formalización de procesos adaptables y permiten un rápido desarrollo y resultados fiables. En estas metodologías la gente es el principal factor de éxito de un proyecto y se propone que exista una buena interacción entre el cliente y el equipo de desarrollo.

2.3.2 Selección de una metodología

El seleccionar una metodología de desarrollo no es una tarea sencilla, ya que se debe tomar en cuenta una serie de factores que intervienen en el desarrollo del software, como el equipo desarrollador, el tipo de sistema requerido, el personal que

utilizará el software, y la forma de trabajo de la dependencia u organización en que se desarrolla.

Para la elección de la metodología utilizada en el desarrollo del sistema descrito en este trabajo, se tomó en cuenta lo siguiente:

- El sistema del PERPAE sería desarrollado por y para la DGAPA, lo que facilitaría establecer una adecuada comunicación entre el equipo desarrollador y los usuarios del sistema.
- Los equipos de desarrollo asignados por la DGAPA para la implementación de sistemas son relativamente pequeños, lo que implica que los integrantes realicen distintas actividades.
- En la DGAPA solo es generada la documentación necesaria durante el desarrollo de los sistemas de información, lo que propicia un desarrollo ágil con resultados continuos.
- La DGAPA no emplea una metodología en específico para desarrollo de software, sino que utiliza técnicas que se basan en la forma de trabajo del equipo de desarrollo.

Debido a lo anterior, y en base al conocimiento que se tiene sobre *Scrum*, se seleccionó esta metodología para el desarrollo del sistema.

Debe de tomarse en cuenta que debido a que la DGAPA no utiliza una metodología en específico, para el desarrollo del sistema del PERPAE se emplearon únicamente las herramientas necesarias y más importantes de *Scrum*, esto con la finalidad de proveer un desarrollo ágil y no entorpecer los procesos y forma de trabajo de la DGAPA.

A continuación se describe la metodología *Scrum*, tomando en cuenta sus principales características y ventajas sobre otras metodologías.

2.3.3 Scrum

Descripción

Scrum es una metodología ágil para desarrollo de software de manera simple, basada en la adaptación continua de las circunstancias del proyecto y orientada a las personas antes que a los procesos.

En *Scrum* es esencial que el equipo de trabajo sea auto-organizado, responsable y autodisciplinado, además debe existir confianza entre los miembros del equipo, propiciando la transparencia y visibilidad en el desarrollo del proyecto.

Es una metodología iterativa incremental, en la que los proyectos se realizan en bloques cortos y fijos llamados *sprints*, que son iteraciones cuya duración va desde semanas hasta un mes de duración. Cada iteración tiene que proporcionar un

resultado completo, o sea, un incremento del producto final que sea susceptible de ser entregado al cliente.

Documentos

Existen elementos utilizados en *Scrum* para controlar el proceso de desarrollo, estos son:

- *Product backlog*: Son los requerimientos que representan las características del sistema que se desean obtener. Es un documento en constante evolución y accesible a todas las personas que intervienen en el desarrollo. Este documento está integrado por los requerimientos funcionales, mejoras y características tecnológicas del software.
- *Sprint backlog*: Documento que contiene la lista de tareas que realizará el equipo durante el *sprint* para generar un incremento. Contiene también, el tiempo estimado y personal responsable de cada tarea.

Historia de usuario

El concepto de “historia de usuario” es comúnmente utilizado en *Scrum* para hacer referencia a los requerimientos del proyecto descritos en lenguaje de usuario.

Proceso de desarrollo

El proceso de desarrollo parte de la lista priorizada de objetivos y requerimientos del producto (*product backlog*), que actúa como plan del proyecto. El ciclo de vida comprende 5 etapas:

1.- Concepto: Fase en la que se crea una visión y alcance del producto o servicio que se quiere obtener y se establece el equipo de trabajo. Es la fase de preparación del proyecto.

2.- Especulación: Después de tener una visión de los que se requiere, el equipo especula y construye hipótesis sobre la información de la visión. Se determinan las limitaciones y la aproximación de lo que se puede producir. Esta etapa se realiza en cada iteración del desarrollo y consiste en:

- Desarrollar y revisar los requerimientos generales del producto.
- Desarrollar una lista con las funcionalidades esperadas.
- Construir un plan de entrega.

Puede incluir también estrategias o planes para la gestión de riesgos.

3.- Exploración: Desarrollo de las funcionalidades del producto en una etapa.

4.- Revisión: El equipo y los usuarios revisan las funcionalidades construidas determinando su dirección y alineación con el objetivo.

5.- Cierre: Obtención del producto esperado y continuidad del mismo.

Reuniones

En *Scrum* se realizan reuniones entre el equipo de trabajo para la planificación de un incremento, en las cuales se determina el trabajo y los objetivos a cumplir. También se realizan reuniones para dar repaso al avance de cada tarea y el trabajo.

Roles

En *Scrum* existen roles que intervienen directamente con el desarrollo del software y a los que se asignan tareas. Estos son:

- Propietario del producto o *product owner*: Representa a los interesados en el producto final y es el responsable del *product backlog*.
- El equipo: Es un equipo multidisciplinar que cubre las habilidades para generar el resultado esperado. Se autogestiona tomando decisiones sobre como realizar su trabajo.
- *Scrum Master*: tiene la responsabilidad del funcionamiento de la metodología.

Existen también otros roles que no forman parte del proceso de *Scrum*, pero están relacionados con el software a desarrollar y son:

- Usuarios finales del software.
- *Stakeholders*: Clientes, vendedores o personal que permite la existencia del software.
- Gerentes o administradores de la organización.

Gráficas

Scrum utiliza algunas herramientas gráficas que ayudan a dar seguimiento al proyecto a lo largo de su ciclo de vida. Estas son:

- Gráfico *Burn-Up*: Herramienta de gestión y seguimiento para el propietario del producto. Presenta las versiones del producto previstas, las funcionalidades de cada una, velocidad estimada, fechas probables para cada versión y avance real.
- Gráfica *Burn-Down*: Herramienta para representar el trabajo de cada sprint.

Ventajas

Ventajas sobre las metodologías tradicionales:

- Es una metodología adaptable a los posibles cambios que surjan durante el desarrollo del sistema.
- Define procesos más flexibles.
- Establece equipos pequeños y organizados.
- Implica únicamente la realización de artefactos y documentación necesaria.
- Establece pocos roles, lo que facilita la comunicación e interacción del equipo de trabajo.
- No es necesaria la realización de un análisis exhaustivo.
- Los integrantes del equipo de trabajo pueden involucrarse en diferentes actividades, lo que propicia el aprendizaje.

Ventajas sobre metodologías basadas en otros ciclos de vida:

- Permite el desarrollo del sistema de forma continua, haciendo entregas constantes, sin depender del término de fases anteriores.
- Permite la detección temprana de errores, lo que disminuye la probabilidad de fallas en el sistema.
- Las iteraciones iniciales, sirven como base en el desarrollo de las iteraciones posteriores.
- Se puede atender los requerimientos con mayor prioridad para los usuarios.
- Permite identificar fácilmente si un proyecto es viable.

2.4 LENGUAJES DE PROGRAMACIÓN

Un lenguaje de programación es un lenguaje artificial utilizado para expresar sentencias que pueden ser interpretadas por una computadora. Está formado por un conjunto de reglas y sentencias que definen la estructura de sus expresiones.

2.4.1 Paradigmas de programación

Un paradigma de programación, es la forma o modelo en que se implementan las estructuras de un lenguaje de programación. Existen varios paradigmas de programación, entre los que se encuentran: programación funcional, programación lógica, programación estructurada, programación orientada a eventos y programación orientada a objetos

A continuación se describe el paradigma de programación orientado a objetos debido a que es uno de los más utilizados para el desarrollo de sistemas de información. Además, es utilizado para el desarrollo del sistema descrito en este trabajo y cuyo uso se justifica posteriormente.

Programación orientada a objetos

La programación orientada a objetos (POO) tiene como propósito desarrollar software capaz de reflejar características del mundo real por medio de objetos con atributos y acciones.

La programación orientada a objetos emplea los siguientes conceptos:

- **Objeto:** Es un concepto real o abstracto que conforma nuestro entorno. En términos de programación, es la instancia de una clase, es decir, una entidad que cuenta con atributos (estructura), acciones (comportamiento) e identidad.
- **Clase:** Es una plantilla que define las propiedades y comportamiento de un tipo de objeto en concreto, es decir, establece los métodos y atributos que tendrá un objeto derivado de la clase. Incluye ciertos principios fundamentales como la abstracción, la herencia, el polimorfismo y el encapsulamiento.
- **Método:** Es un algoritmo asociado a una clase cuya ejecución se produce tras la recepción de un mensaje.
- **Abstracción:** Es la adaptación de un objeto con las propiedades necesarias, excluyendo aquellas que no lo son, con la finalidad de obtener un objeto con cualidades útiles para el desarrollo del software.
- **Herencia:** Es la propiedad de una clase de poseer las características (atributos y acciones) de una clase de la que hereda.

- Polimorfismo: Es la propiedad de un método de comportarse de manera diferente dependiendo del contexto en que se encuentre. Es útil cuando diferentes objetos pueden realizar la misma acción pero de manera distinta.
- Sobrecarga: Consiste en utilizar el mismo nombre para métodos o clases distintas. Es un tipo de polimorfismo.
- Encapsulamiento: Es la propiedad de reunir todos los elementos que pertenecen a una entidad, es decir atributos y métodos, en el mismo nivel de abstracción. En otro concepto, es llamado también encapsulamiento a la propiedad de ocultamiento, que es la propiedad de un objeto de ocultar ciertas propiedades que no requieren ser “vistas” por otros objetos o por el mundo exterior.
- Modularidad: Es la propiedad que permite dividir una aplicación en módulos o partes más pequeñas que deben ser independientes de la aplicación y de los demás módulos.
- Asociaciones: La asociación entre clases u objetos, se produce cuando estos requieren trabajar en conjunto. La multiplicidad o diversificación indica la cantidad de objetos de una clase que se relacionan con otro objeto de una clase asociada.

Existen muchas ventajas para utilizar la programación orientada a objetos, algunas son:

- Permite la reutilización y extensión de código, y el desarrollo de sistemas complejos.
- Relaciona el sistema al mundo real.
- Facilita el desarrollo y mantenimiento del software.
- Permite crear aplicaciones más estables, flexibles y uniformes.
- Facilita la documentación y el diseño.

La programación orientada a objetos representa una forma muy particular de desarrollar software, por lo que su empleo tiene algunas desventajas como:

- Puede resultar complicado adaptarse a este paradigma de programación.
- La ejecución de un software orientado a objetos es más lenta.
- Puede emplearse mayor cantidad de código.
- La depuración de código puede resultar más compleja, ya que se necesita recorrer el árbol de herencia para encontrar un error.

2.5 LENGUAJE UNIFICADO DE MODELADO

El Lenguaje Unificado de Modelado o *Unified Modeling Language* (UML) es el estándar más utilizado de la OMG (*Object Management Group*) para modelar, especificar, documentar, detallar y construir sistemas de software. UML permite describir aspectos conceptuales como procesos de negocio, funciones del sistema, aspectos de programación, esquemas de base de datos, y componentes del sistema.

Los objetivos principales de UML son:

- Ser un lenguaje universal para modelado, no solo sistemas de software, si no cualquier tipo de sistemas.
- Ser un lenguaje simple.
- Establecer una relación entre los conceptos y los artefactos ejecutables.
- Ayudar a manejar problemas de sistemas complejos.
- Permitir especificar las características de un sistema.
- Apoyar en la documentación de los proyectos.

2.5.1 Diagramas

UML ofrece distintos tipos de diagramas utilizados para modelar sistemas y presentar diversas perspectivas de los mismos. A continuación se presenta una breve descripción de cada uno de los diagramas que ofrece UML.

Diagramas de clases

Son utilizados para modelar la estructura estática de las clases de objetos en el sistema. Es el diagrama central del diseño de un sistema orientado a objetos.

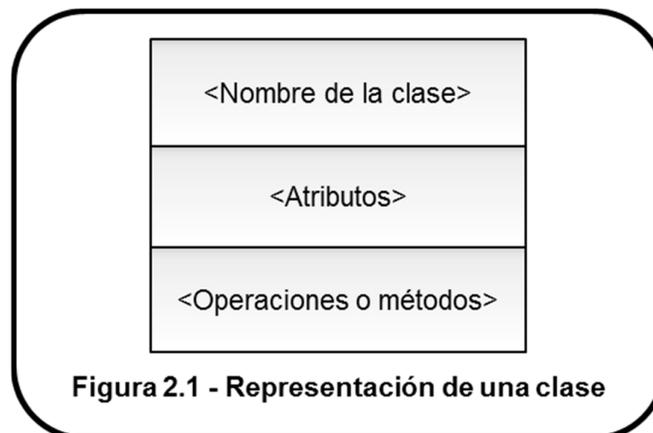
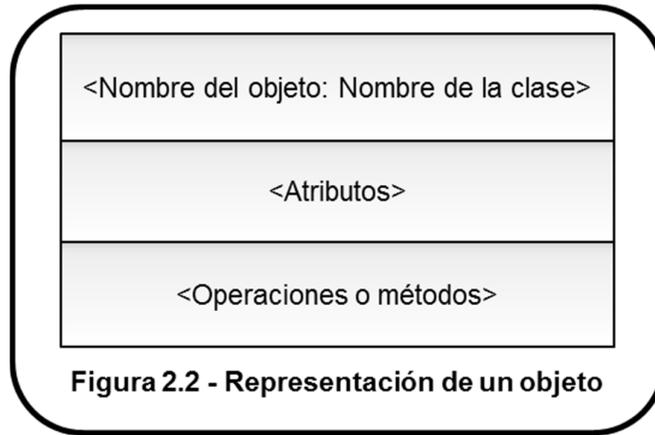


Figura 2.1 - Representación de una clase

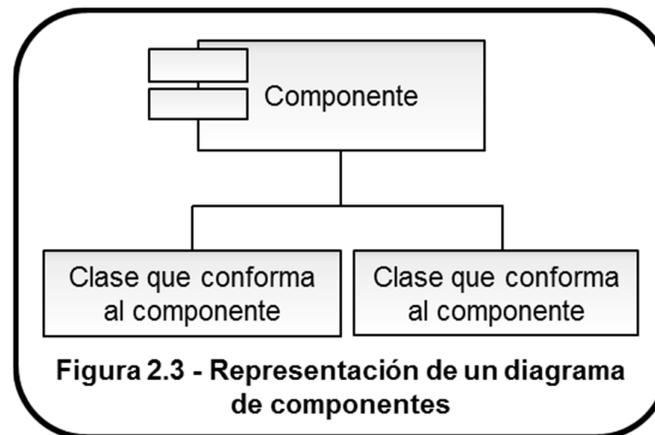
Diagramas de objetos

Son los diagramas utilizados para modelar la estructura estática de los objetos en el sistema.



Diagramas de componentes

Son diagramas que representan elementos físicos, interfaces y relaciones de un sistema.



Diagramas de implementación o despliegue

Son los diagramas utilizados para modelar la distribución general de hardware del sistema.

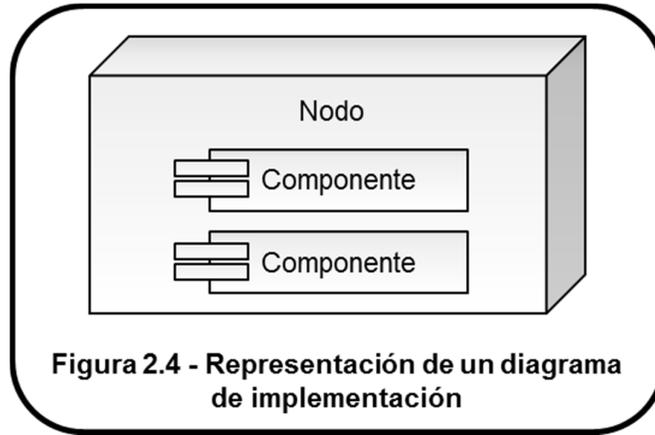


Figura 2.4 - Representación de un diagrama de implementación

Diagramas de paquetes

Son los diagramas utilizados para representar de forma jerárquica, la estructura lógica y las relaciones entre los componentes del software. Permiten organizar diagramas, como los de clases y caso de uso, en el mismo espacio de nombre.

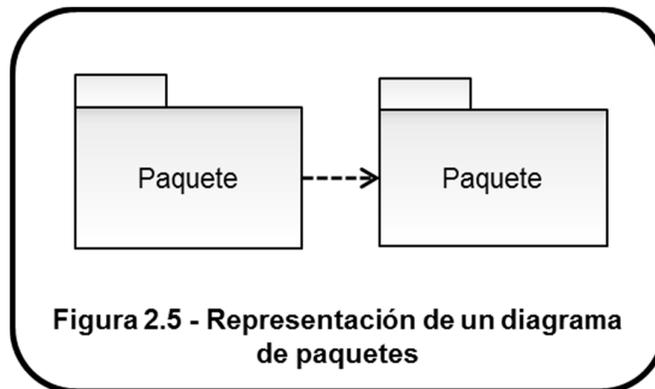


Figura 2.5 - Representación de un diagrama de paquetes

Diagramas de estado

Son utilizados para modelar el comportamiento dinámico de los objetos en el sistema en tiempo real.

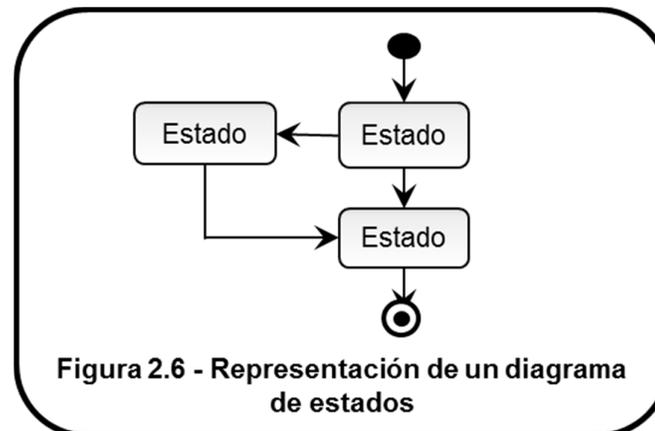
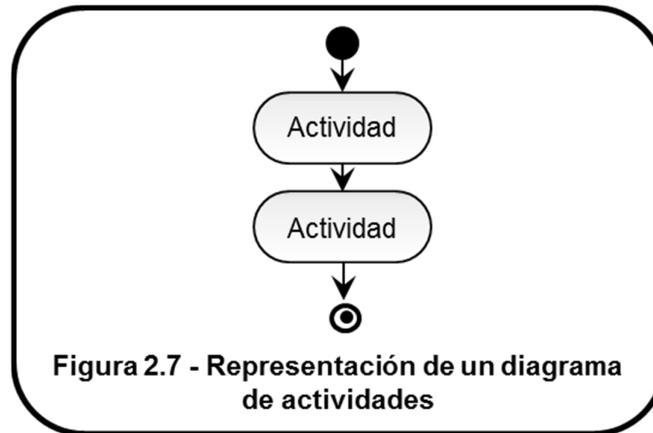


Figura 2.6 - Representación de un diagrama de estados

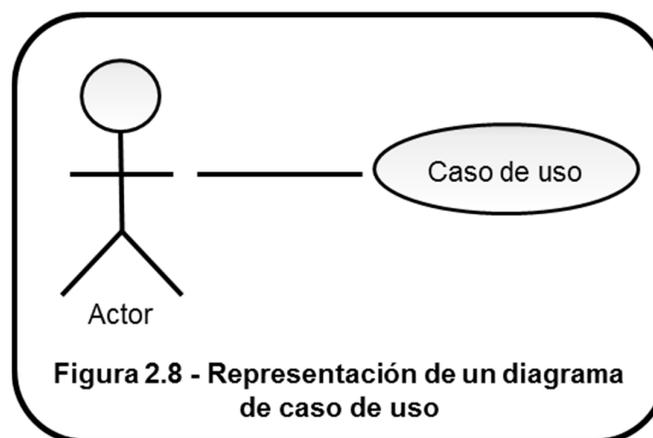
Diagramas de actividad

Se utilizan para modelar el comportamiento de los casos de uso, objetos u operaciones de un sistema. Muestran una visión de las actividades que ocurren durante una operación o proceso.



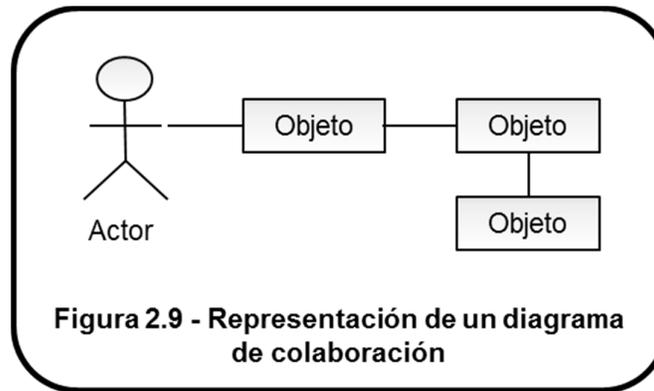
Diagramas de caso de uso

Son los diagramas utilizados para analizar y modelar los requerimientos del sistema y la problemática a resolver desde la perspectiva del usuario. Se componen de actores y casos de uso. Los actores son los usuarios o elementos que interactúan con el sistema y los casos de uso representan la forma en la que se comporta el sistema y los escenarios en que se puede encontrar.



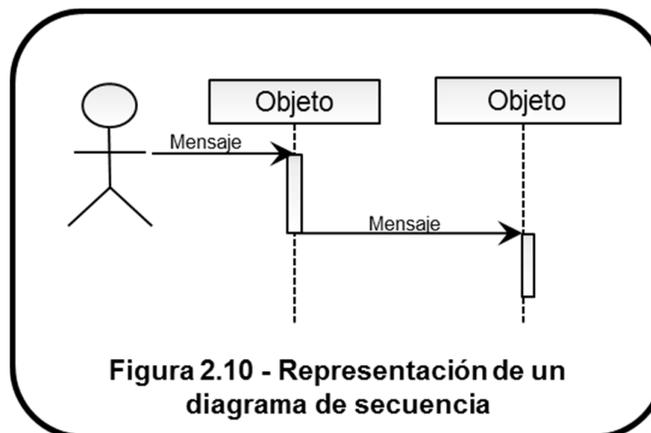
Diagramas de comunicación o colaboración

Son utilizados para modelar interacciones y relaciones entre objetos del sistema. Son una extensión de los diagramas de objetos, centrados en estudiar todos los efectos que puede producir un objeto en un escenario.



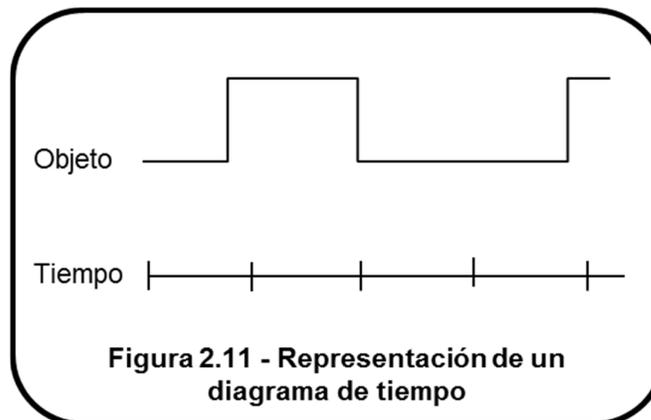
Diagramas de secuencia

Sirven para modelar el flujo de información entre componentes de un sistema. Se generan en base de los casos de uso y contienen detalles sobre la implementación de cada escenario, incluyendo los objetos, clases y mensajes.



Diagramas de tiempo

Son los diagramas utilizados para representar el cambio de estado de elementos del sistema a lo largo del tiempo. Pueden representarse también las interacciones entre eventos, restricciones y duración de tiempos.



2.6 BASES DE DATOS

Una base de datos, es un conjunto de datos relacionados y almacenados ordenadamente para su uso. Se compone de hardware, que es el medio físico en el que se almacenan los datos; software, que es el sistema administrador de la base de datos; y los datos.

Existen diferentes modelos de bases de datos de acuerdo a su forma de administrar la información. Entre los modelos más comunes se encuentran: jerárquico, de red, transaccional, multidimensional, orientado a objetos, deductivo, lógico y relacional.

2.6.1 Sistema gestor de base de datos

Es una herramienta tecnológica específica que funciona como interfaz entre la base de datos y las aplicaciones que la utilizan, permite almacenar y acceder a los datos de forma rápida. Se compone de los siguientes elementos:

- Motor o núcleo: Convierte en operaciones de lectura y escritura los requerimientos lógicos de entrada y salida.
- Herramientas de definición: Permiten modificar y definir la estructura o esquema de la base de datos.
- Interfaz de procesamiento: Provee las funciones de actualización, despliegue y visualización de los datos.
- Desarrollo de aplicaciones: Permite la generación de aplicaciones.
- Diccionario de datos: Es la descripción de la estructura y relaciones de la base de datos.

Algunos de los gestores de bases de datos más comunes son: *MySql, dBase, IBM DB2, IBM Informix, Microsoft Access, Oracle, Microsoft SQL Server, PostgreSQL.*

2.6.2 Características

Existen características que debe poseer una base de datos para que sea una herramienta adecuada de gestión de información. Estas características son:

- Proporcionar integridad e independencia a los datos almacenados.
- Minimizar la redundancia en los datos.
- Controlar la concurrencia, es decir, permitir el acceso a varios usuarios a un objeto de datos al mismo tiempo.

- Permitir recuperar y restaurar la base de datos.
- Permitir mantener los datos seguros, es decir, resguardados y con claves de acceso solo a usuarios autorizados.
- Provee recursos para establecer relaciones y restricciones entre los datos.

2.6.3 El lenguaje estructurado de consulta o *Structured Query Language* (SQL)

Es un lenguaje declarativo, utilizado para realizar operaciones y consultas sobre una base de datos. Es el lenguaje por excelencia de los sistemas de gestión de las bases de datos relacionales y se divide en:

- DDL (*Data Definition Language* o lenguaje de definición de datos): Es el lenguaje utilizado para la modificación de la estructura de la base de datos. Permite crear, modificar y borrar las estructuras de la base de datos. Utiliza sentencias como: *create*, *drop*, *alter* y *truncate*.
- DML (*Data Manipulation Language* o lenguaje de manipulación de datos): Es el lenguaje que permite la manipulación o consulta de datos. Utiliza sentencias como: *select*, *insert*, *update* y *delete*.
- DCL (*Data Control Language* o lenguaje de control de acceso de datos): Permite controlar el acceso a los datos de una base de datos. Utiliza sentencias como: *grant*, *revoke*, *commit* y *rollback*.

2.6.4 Modelado de datos de una base de datos relacional

Un modelo de datos es una representación de la base de datos, que puede incluir las estructuras, restricciones de integridad y operaciones de la misma. Existen los siguientes modelos para modelar una base de datos:

- Modelo conceptual: Permite representar de forma descriptiva, los datos que intervienen en la base de datos. El modelo entidad relación, es el modelo más común de los modelos conceptuales, y permite representar las entidades de un sistema, así como sus propiedades o atributos, y las relaciones entre ellas.
- Modelo lógico: Son modelos utilizados para representar una base de datos de manera operacional, por medio de tablas, relaciones y atributos. Para el diseño de este modelo, comúnmente se utiliza la normalización, que es un conjunto de reglas para proveen integridad y permiten evitar la redundancia en una base de datos.
- Modelo físico: Son las tablas con sus características, organización y estructura de almacenamiento representadas a nivel del manejador de la base de datos.

CAPÍTULO III

PREPARACIÓN

En este capítulo se describen los conceptos y actividades correspondientes a la fase de preparación del sistema de información del PERPAE en base a la metodología seleccionada.

Cabe señalar que en este capítulo se presenta un análisis general de información que sirvió de base para determinar el *product backlog*, el cual, fue requerido para guiar el desarrollo del sistema.

3.1 DEFINICIÓN DEL SISTEMA

3.1.1 El caso de negocio

El sistema se sitúa en el contexto académico de la UNAM, en específico, en el ámbito administrativo de la DGAPA.

El sistema está basado en la metodología de desarrollo *Scrum* y es un sistema institucional y específico de apoyo a nivel de conocimiento en la administración de información relacionada con el PERPAE.

Los usuarios del sistema, que actúan como clientes del mismo, pertenecen a la Subdirección de Estímulos y Reconocimientos de la DGAPA.

Los beneficios obtenidos con el desarrollo del sistema son de tipo no financiero, ya que al ser un software elaborado por y para la DGAPA, el beneficio principal es el apoyo en las actividades de gestión de información del PERPAE.

Para el desarrollo del sistema, son empleados recursos de la DGAPA, tanto humanos como de infraestructura.

3.1.2 Objetivos

Los principales objetivos planteados para el sistema son:

- Apoyar en la administración de información relacionada con el PERPAE.
- Ser una herramienta adaptable y gestionable tecnológicamente por la DGAPA.

3.1.3 Visión

La visión planteada para el sistema es:

El sistema a desarrollar será una herramienta funcional, útil, eficaz, gestionable y adaptable, cuyo propósito será apoyar en la administración de información del PERPAE.

3.1.4 Definición del equipo de trabajo

Para el desarrollo del sistema del PERPAE, se asignó un equipo de trabajo conformado por personal capacitado cuya función fue apoyar en diferentes tareas relacionadas con la ingeniería de software. Este equipo de trabajo fue conformado por:

- Ing. Arturo Bahena Armas – Jefe del Departamento de Estadísticas y Sistemas.
- Ing. Víctor Manuel Viguera Sánchez – Jefe de área del Departamento de Estadísticas y Sistemas.
- Enrique Israel Salazar Migueles – Asistente de procesos del Departamento de Estadísticas y Sistemas.

El equipo de desarrollo mencionado, es parte de la Subdirección de Sistemas, Diagnóstico e Información Académica de la DGAPA.

Además del equipo de desarrollo, se estableció el siguiente grupo de usuarios que apoyaron en distintas actividades durante el ciclo de vida del sistema y que forman parte de la Subdirección de Estímulos y Reconocimientos, encargada de gestionar el PERPAE:

- Mtra. María Lilia Rosas García – Jefa del Departamento de Reconocimientos.
- Lic. Jessica Castañeda García - Asistente de procesos del Departamento de Reconocimientos.

3.1.5 Definición de terminado

Para el desarrollo del sistema, se definió como terminada una historia, cuando los requerimientos involucrados fueran cubiertos en su totalidad.

El definir las condiciones de terminado para una historia, permitió darla por concluida y proceder a cubrir los requerimientos de la historia siguiente.

3.2 PRODUCT BACKLOG DEL SISTEMA

El *product backlog* reúne los requerimientos funcionales, características tecnológicas y mejoras, que sirven de base para el desarrollo del sistema de información. Además, es un documento de apoyo en la planeación y control del sistema, por tal motivo, fue elaborado en esta fase.

Para la elaboración del *product backlog*, se recopiló y analizó información relacionada con el PERPAE y con las tecnologías utilizadas en la DGAPA. Esto, con la finalidad de establecer de manera general, las características con que debía contar el sistema de información.

3.2.1 Recopilación y análisis general de información

La información relacionada al PERPAE, fue proporcionada por los usuarios del sistema, ya que es el personal que tiene mayor conocimiento al respecto. Esta información fue proporcionada a través de entrevistas y por medio de documentos con información detallada sobre el PERPAE.

Por otra parte, se recopiló información para determinar las características con las que debía contar el sistema de información para que fuera adaptable a las tecnologías de la DGAPA. Esta información fue proporcionada por el Departamento de Estadísticas y Sistemas de la Subdirección de Sistemas, Diagnóstico e Información Académica a través de entrevistas.

Con la recopilación de información, se determinaron los procesos, personas y elementos tecnológicos relacionados con el PERPAE y que se describen a continuación.

Procesos involucrados con el PERPAE

- Registro de académicos en el PERPAE: Proceso que se lleva cabo cuando un académico es designado emérito y consiste en darlo de alta en el PERPAE.
- Administración de información de eméritos: Proceso que involucra el registro, seguimiento y consulta de la siguiente información de eméritos:
 - Datos personales.
 - Datos referentes al fallecimiento y estado de salud.
 - Domicilios.
 - Contactos que incluyen cónyuge, discípulos y asistentes.
 - Situación laboral frente a la UNAM.
 - Designaciones.
 - Distinciones recibidas.
 - Actividades de difusión de obra y pensamiento.

- Estudios y nivel académico.
- Cuenta bancaria.
- Seguros de vida y gastos médicos.
- Padecimientos.
- Otorgar apoyo económico a eméritos: Proceso que consiste en determinar las condiciones del apoyo económico que es otorgado a eméritos.
- Otorgar seguro de gastos médicos y seguro de vida a eméritos: Proceso que consiste en otorgar un seguro de vida y un seguro de gastos médicos a eméritos en base a su situación personal.
- Difundir obra y pensamiento de eméritos: Proceso que involucra administrar actividades para la difusión de pensamiento y obra de eméritos.
- Generar reportes y gráficas: Proceso que involucra la generación de información estadística y general sobre eméritos, en forma de gráficas y reportes.

Personas involucradas con el PERPAE

Se identificaron los siguientes roles involucrados en los procesos del PERPAE:

- Personal administrativo: Personal capacitado y autorizado para registrar, consultar y dar seguimiento a información del PERPAE.
- Personal de consulta: Personal autorizado para consultar información estadística y general del PERPAE.

Sistemas cooperativos

Se identificaron los siguientes elementos tecnológicos que tienen relación con el PERPAE.

- GeDGAPA: Sistema web de administración de información que concentra sistemas de distintos programas y servicios de la DGAPA.
- Núcleo: Esquema de base de datos en *PostgreSQL* que administra información de personal de la UNAM.
- Catálogos: Esquema de base de datos en *PostgreSQL* que administra catálogos de información, utilizados en distintos programas de la DGAPA y relacionados al personal académico.
- Nómina: Esquema de base de datos en *PostgreSQL* que contiene información relacionada con el personal académico que labora en la UNAM.

Características requeridas

En base a la información que fue recopilada, se definieron las siguientes características requeridas por el sistema de información:

- Accesible y seguro: El sistema debe ser utilizado solo por el personal autorizado y autenticado, y debe permitir el manejo de información de una forma confiable.
- Amigable e intuitivo: Fácil en su manejo y con un flujo de páginas sencillo.
- Fácil en su mantenimiento: Actualizable y escalable sencillamente.

Debido al tipo de información que se maneja en el PERPAE, el sistema requiere contar con lo siguiente:

- Un mecanismo de alertas para indicar si existe información referente a eméritos no actualizada o próxima a expirar.
- Una bitácora de control para registrar los movimientos realizados por los usuarios del sistema.

También se determinaron las siguientes características para adaptar el sistema de información a las tecnologías de la DGAPA:

- El sistema debe ser una aplicación web debido a que requiere ser integrado a un sistema de este tipo, y porque permite portabilidad, compatibilidad y disponibilidad.
- El sistema debe ser desarrollado en lenguaje *Java* debido a que requiere ser compatible con el sistema al que será integrado. Por otra parte, el lenguaje *Java*, es un estándar de programación utilizado en muchas tecnologías, lo que permite que el sistema sea escalable y adaptable a distintos entornos.
- Debe contar con una base de datos en *PostgreSQL* debido a que requiere ser integrado con esquemas de base de datos que utilizan esta herramienta.

3.2.2 Definición del *product backlog*

Una vez analizada la información del sistema, se definió el *product backlog*, que en *Scrum*, es el documento más importante debido a que dirige la construcción del sistema.

Se debe tomar en cuenta que el *product backlog* es un documento que está siempre en evolución debido a que, siendo *Scrum* una metodología flexible, los requerimientos pueden cambiar durante el desarrollo del sistema. Por tal motivo, se le debe dar el apropiado seguimiento para asegurar que los requerimientos son definidos de una manera adecuada y estén siempre actualizados.

Con el fin de presentar al lector información clara, se muestra a continuación el *product backlog* que resultó de la definición total y precisa de los requerimientos del sistema después de pasar por una serie de modificaciones.

Es importante mencionar que los requerimientos presentados en el *product backlog*, son detallados en los próximos capítulos de este trabajo.

Los datos con que cuenta cada requerimiento del *product backlog* generado para el sistema son:

- Identificador del requerimiento.
- Descripción del requerimiento.
- Tipo de requerimiento (funcional o no funcional).
- Estimación o número de días de trabajo designados para el desarrollo del requerimiento.
- *Sprint* o número de iteración para cubrir el requerimiento. Se debe de tomar en cuenta que la duración de los *sprints* para el sistema a desarrollar tienen una duración de 20 días de trabajo.

Id	Descripción	Tipo de requerimiento	Estimación total	Sprint
R01	Ser una herramienta web desarrollada en lenguaje <i>Java</i>	No funcional	5	1
R02	Contar con una base de datos en <i>PostgreSQL</i>	No funcional	15	1
R03	Autenticar usuarios	Funcional	2	2
R04	Generar un menú de opciones para cada tipo de usuario	Funcional	2	2
R05	Permitir la administración de eméritos	Funcional	5	2
R06	Permitir la administración de información personal de eméritos	Funcional	4	2
R07	Permitir la administración de información del estado de salud de eméritos	Funcional	3	2
R08	Permitir la administración de información de domicilios de eméritos	Funcional	3	3
R09	Permitir la administración de información de contactos de eméritos	Funcional	4	3
R10	Permitir la administración de información de la situación laboral de eméritos	Funcional	3	3
R11	Permitir la administración de información de designaciones otorgadas a eméritos	Funcional	3	3
R12	Permitir la administración de un catálogo de distinciones	Funcional	2	3

Id	Descripción	Tipo de requerimiento	Estimación total	Sprint
R13	Permitir la administración de un catálogo de actividades de difusión	Funcional	2	4
R14	Permitir la administración de un catálogo de pólizas	Funcional	2	4
R15	Permitir la administración de un catálogo de compañías bancarias y de seguros	Funcional	2	4
R16	Permitir la administración de pólizas contratadas	Funcional	3	4
R17	Permitir la administración de información de actividades de difusión en que participan eméritos	Funcional	3	4
R18	Permitir la administración de información de distinciones de eméritos	Funcional	3	4
R19	Permitir la administración de información de estudios realizados por eméritos	Funcional	3	5
R20	Permitir la administración de información de cuenta bancaria de eméritos	Funcional	3	5
R21	Permitir la administración de información de pólizas de seguros de eméritos	Funcional	5	5
R22	Permitir la administración de información de padecimientos de eméritos	Funcional	5	5
R23	Generar alertas para notificar la información no actualizada o próxima a expirar	Funcional	2	6
R24	Permitir la generación de reportes de información de eméritos	Funcional	8	6
R25	Permitir la generación de gráficas de información de eméritos	Funcional	8	6
R26	Permitir su integración al sistema GeDGAPA	No funcional	2	6
R27	Proveer un fácil mantenimiento	No funcional	4	2,3,4,5
R28	Registrar los movimientos realizados por los usuarios en una bitácora	Funcional	4	2,3,4,5
R29	Contar con una interfaz gráfica amigable	No funcional	4	2,3,4,5
R30	Proveer seguridad de la información	No funcional	4	2,3,4,5

Tabla 3.1 – Product backlog.

3.3 DEFINICIÓN DE ACTIVIDADES y PLAN DE ENTREGA

3.3.1 Definición de actividades

Dentro de las actividades de preparación, se definieron las tareas a realizar para cubrir cada uno de los requerimientos del sistema en base al *product backlog*. A continuación se presenta una tabla con las diferentes actividades definidas y que son presentadas a detalle en los siguientes capítulos de este trabajo:

Actividad	Responsable(s)
Análisis de información	Sandra Lilibiana Ramírez Mora, Ing. Víctor Manuel Viguera Sánchez
Modelado de datos	Sandra Lilibiana Ramírez Mora, Ing. Arturo Bahena Armas
Elaboración de casos de uso	Sandra Lilibiana Ramírez Mora
Diseño de interfaz gráfica	Sandra Lilibiana Ramírez Mora
Diseño de reportes y gráficas	Sandra Lilibiana Ramírez Mora
Implementación de base de datos	Sandra Lilibiana Ramírez Mora, Ing. Arturo Bahena Armas
Generación y configuración del proyecto	Sandra Lilibiana Ramírez Mora, Enrique Israel Salazar Migueles
Codificación	Sandra Lilibiana Ramírez Mora
Generación de reportes y gráficas	Sandra Lilibiana Ramírez Mora
Pruebas	Sandra Lilibiana Ramírez Mora, Ing. Arturo Bahena Armas, Ing. Víctor Manuel Viguera
Implantación del sistema	Sandra Lilibiana Ramírez Mora, Enrique Israel Salazar Migueles

Tabla 3.2 – Actividades y responsables.

3.3.2 Plan de entrega

En base al *product backlog* generado, se estimó que el sistema sería desarrollado en un tiempo de 118 días de trabajo, equivalente a 6 meses. Esto, sin tomar en cuenta el tiempo que se empleó para realizar el *product backlog*, que fue de tres semanas.

Por otra parte, se establecieron entregas constantes durante el desarrollo del sistema con la finalidad de presentar a los *product owners* o usuarios del sistema, el avance e incrementos generados en un periodo de tiempo.

CAPÍTULO IV

ANÁLISIS Y DISEÑO

En este capítulo se presentan las actividades referentes al análisis y diseño realizadas durante el ciclo de vida de desarrollo del sistema. La importancia de realizar dichas actividades radica en que indican lo que se requiere desarrollar y como debe realizarse, permitiendo contestar preguntas como:

¿Qué información se requiere gestionar?

¿Cómo debe gestionarse la información en cuestión?

¿Cómo deben implementarse los componentes del sistema?

4.1 ANÁLISIS DE REQUERIMIENTOS

El análisis detallado de los requerimientos funcionales permitió definir los procesos que el sistema requería realizar, asimismo, facilitó el análisis de la información relacionada y el diseño del modelo de datos.

El análisis de los requerimientos no funcionales, permitió describir las características tecnológicas del sistema, las cuales, fueron tomadas en cuenta para el diseño y desarrollo del mismo.

A continuación se presenta la descripción detallada de los requerimientos funcionales y no funcionales del sistema, los cuales fueron presentados de manera general en el capítulo anterior mediante el *product backlog*.

4.1.1 Requerimientos funcionales

Autenticar usuarios

Para ingresar al sistema, los usuarios deben autenticarse por medio de un nombre de usuario y una contraseña previamente asignados por la DGAPA, lo que permite controlar el acceso al sistema y proveer seguridad en el mismo.

Generar un menú de opciones para cada tipo de usuario

Debido a que los usuarios administradores y los usuarios de consulta están autorizados para realizar diferentes actividades, el sistema debe presentar un menú de opciones diferente para cada tipo de usuario. Mientras que los usuarios administradores están facultados para la realización de todos los procesos relacionados con el PERPAE, los usuarios de consulta únicamente pueden realizar consultas de información por medio de gráficas y reportes.

Permitir la administración de eméritos

El sistema debe permitir a usuarios administradores el alta, baja y búsqueda de eméritos en el sistema.

La búsqueda de eméritos debe poder realizarse mediante:

- Registro Federal de Contribuyentes (RFC)
- Apellido paterno
- Número de expediente

Permitir la administración de información personal de eméritos

El sistema debe permitir a usuarios administradores el registro y actualización de datos personales de eméritos, tales como:

- Registro Federal de Contribuyentes (RFC)

- Número de expediente que otorga la UNAM.
- Nombre(s).
- Apellido paterno.
- Apellido materno.
- Clave Única de Registro de Población (CURP).
- Estado civil.
- Lugar específico de nacimiento.
- País de nacimiento.
- Nacionalidad.
- Fecha de nacimiento.
- Edad.
- Género.
- Correo electrónico.
- Observaciones o notas correspondientes.

Permitir la administración de información de la situación personal de eméritos

El sistema debe permitir a usuarios administradores el alta y baja de datos relacionados a la situación personal de eméritos, que incluye estado de salud y/o datos de fallecimiento. También debe permitir visualizar el historial de la situación personal de eméritos.

La administración de este tipo de información es importante ya que permite llevar un seguimiento del estado personal de eméritos y poder generar estadísticas y reportes.

Los datos referentes a la situación personal son:

- Estado o situación.
- Fecha del estado o situación.
- Observaciones o notas correspondientes.

Permitir la administración de información de domicilios de eméritos

El sistema debe permitir a usuarios administradores el alta, baja, actualización y consulta de datos referentes a domicilios de eméritos con el fin de tener una referencia para localizarlos cuando sea requerido y enviarles correspondencia.

Los datos en relación a domicilios son:

- Tipo de domicilio.

- País.
- Calle.
- Colonia.
- Delegación.
- Ciudad.
- Código postal.
- Teléfono(s).
- Fax.
- Observaciones o notas correspondientes.

Permitir la administración de información de contactos de eméritos

El sistema debe permitir a usuarios administradores el alta, baja, actualización y consulta de contactos personales de eméritos, que incluyen cónyuge, discípulos y asistentes. Esta información es importante porque sirve como referencia para localizar a eméritos o pedir referencias sobre los mismos en un momento dado.

La información sobre contactos incluye:

- Tipo de contacto
- Datos personales
 - Registro Federal de Contribuyentes (RFC).
 - Género.
 - Nombre(s).
 - Apellido paterno.
 - Apellido Materno.
 - Fecha de nacimiento
 - Fecha de fallecimiento
- Datos de nivel académico
 - Grado o nivel académico.
 - Título obtenido.
 - Abreviatura del título obtenido.
- Datos de domicilio
 - Tipo de domicilio o teléfono.
 - Teléfono(s).
 - País.

- Ciudad.
- Calle.
- Colonia.
- Delegación.
- Fax.
- Correo electrónico.
- Observaciones o notas correspondientes.

Permitir la administración de información de la situación laboral de eméritos

Con la finalidad de llevar un seguimiento de la situación de eméritos frente a la UNAM, el sistema debe permitir a usuarios administradores el registro y actualización de datos laborales de eméritos tales como:

- Años de antigüedad.
- Fecha de ingreso a la UNAM.
- Ley del ISSSTE relacionada.
- Tipo de horas de asignatura impartidas.
- Número de horas de asignatura impartidas.
- Datos de jubilación:
 - Fecha de jubilación.
 - Tipo de horas de asignatura impartidas al momento de la jubilación.
 - Número de horas de asignatura impartidas al momento de la jubilación.
 - Nombramiento al momento de la jubilación.
 - Dependencia de adscripción al momento de la jubilación.
 - Situación académica posterior a la jubilación.
 - Observaciones o notas correspondientes.

La gestión de este tipo de información también es importante para la generación de reportes y gráficas.

Permitir la administración de información de designaciones otorgadas a eméritos

El sistema debe permitir a usuarios administradores el alta, baja, actualización y consulta de los siguientes datos referentes a designaciones otorgadas a eméritos:

- Designación otorgada.
- Dependencia que otorga la designación.
- Subsistema al que pertenece la dependencia que otorga la designación.

- Fecha de designación.
- Nombramiento del emérito al recibir la designación.
- Dependencia de adscripción del emérito al recibir la designación.

Esta información resulta de gran importancia debido a que incluye los datos del momento en que un académico es designado emérito, y por lo tal, origina su ingreso al PERPAE. Además, es indispensable para la generación de reportes y gráficas.

Permitir la administración de información de actividades de difusión en que participan eméritos

Debido a que la realización de actividades de difusión de obra y pensamiento de eméritos es uno de los objetivos del PERPAE, la gestión de este tipo de información es una actividad importante.

El sistema debe permitir a usuarios administradores el alta, baja, actualización y consulta de los siguientes datos referentes a actividades de difusión en que participan los eméritos:

- Elemento o actividad de difusión.
- Tipo de rol desempeñado por eméritos en la actividad de difusión.
- Descripción de la actividad.
- Fecha en que se realiza la actividad.
- País en que se realiza la actividad.
- Lugar específico donde se realiza la actividad.
- Observaciones o notas correspondientes.

Permitir la administración de información de distinciones otorgadas a eméritos

El sistema debe permitir a usuarios administradores el alta, baja, actualización y consulta datos referentes a distinciones otorgadas a eméritos por la UNAM u otras instituciones.

Los datos referentes a distinciones incluyen:

- País donde es otorgada la distinción.
- Ámbito de la distinción.
- Institución que otorga la distinción.
- Nombre de la distinción.
- Año en que se otorga la distinción.
- Observaciones o notas correspondientes.

Permitir la administración de información de estudios realizados por eméritos

El sistema debe permitir a usuarios administradores el alta, baja, actualización y consulta de los siguientes datos referentes a estudios realizados por eméritos:

- País donde se realizó el estudio.
- Institución donde se realizó el estudio.
- Grado académico obtenido.
- Título obtenido.
- Abreviatura del título obtenido.
- Año de titulación.
- Disciplina del estudio.
- Área de la disciplina del estudio.
- Especialidad del estudio.
- Área de especialidad del estudio.
- Indicador del estudio máximo del emérito.
- Observaciones o notas correspondientes.

La gestión de este tipo de información proporciona una referencia de los niveles académicos obtenidos por eméritos.

Permitir la administración de información de la cuenta bancaria de eméritos

El sistema debe permitir a usuarios administradores el registro y actualización de los siguientes datos referentes a la cuenta bancaria donde es otorgado el apoyo económico a eméritos:

- Nombre de la compañía bancaria.
- Número de sucursal.
- Tipo de estímulo o pago otorgado.
- Número de cuenta.
- Monto otorgado.
- Observaciones o notas correspondientes.

Permitir la administración de información de pólizas de seguros de eméritos

El sistema debe permitir a usuarios administradores el alta, baja, actualización y consulta de información referente a pólizas de seguros de vida y gastos médicos que la UNAM otorga al personal académico emérito.

Los datos a administrar en relación a pólizas de seguros de vida son:

- Nombre de la póliza.
- Compañía con que se contrata la póliza.
- Vigencia de la póliza.
- Monto o suma asegurada.
- Datos de beneficiarios:
 - Nombre.
 - Parentesco.
 - Porcentaje.
 - Observaciones.
- Fecha de presentación.
- Fecha de recepción del emérito.
- Fecha de recepción de la dependencia.
- Observaciones o notas correspondientes.

Los datos a administrar en relación a pólizas de seguros de gastos médicos son:

- Nombre de la póliza.
- Compañía con que se contrata la póliza.
- Vigencia de la póliza.
- Monto o suma asegurada.
- Datos de beneficiarios:
 - Nombre.
 - Fecha de nacimiento.
 - Fecha desde que es beneficiario de la póliza.
- Fecha de recepción del emérito.
- Número de póliza.
- Fecha de antigüedad del titular.
- Monto deducible.
- Monto del coaseguro.
- Observaciones o notas correspondientes.

Permitir la administración de información de padecimientos de eméritos

El sistema debe permitir a usuarios administradores el alta, baja, actualización y consulta de datos relacionados con padecimientos de eméritos, los cuales, son financiados por el seguro de gastos médicos de cada emérito.

Debido a que los padecimientos generan movimientos, como operaciones o consultas, el sistema debe permitir el alta, baja, consulta y actualización de dichos movimientos.

Los datos referentes a padecimientos que requieren ser administrados son:

- Nombre del padecimiento.
- Monto con que se cuenta para atender el padecimiento.
- Fecha de inicio del padecimiento.
- Monto total deducible.
- Monto total del coaseguro.
- Monto disponible.

Por otra parte, para la administración de movimientos de padecimientos es requerido gestionar los siguientes datos:

- Nombre del movimiento.
- Número de reclamación.
- Póliza relacionada.
- Tipo de pago.
- Fecha presentación de eméritos.
- Fecha de depósito.
- Monto reclamado.
- Monto no cubierto.
- Monto procedente.
- Monto deducible
- Monto del coaseguro.
- Monto total.
- Observaciones o notas correspondientes.

Permitir la administración de un catálogo de distinciones

El sistema debe permitir a usuarios administradores el alta y baja de información referente a las distinciones que pueden ser otorgadas a eméritos.

Los datos en relación a las distinciones son:

- Nombre de la distinción.
- Abreviatura o siglas de la distinción.

Permitir la administración de un catálogo de actividades de difusión

El sistema debe permitir a usuarios administradores el alta y baja de los elementos que permiten administrar información sobre actividades de difusión de eméritos.

Permitir la administración de un catálogo de pólizas

El sistema debe permitir a usuarios administradores el alta y baja de información referente a las pólizas con que pueden contar eméritos.

Los datos en relación a las pólizas son:

- Nombre de la póliza.
- Tipo de póliza.

Permitir la administración de un catálogo de compañías bancarias y aseguradoras

El sistema debe permitir a usuarios administradores el alta y baja de información referente a las compañías contratadas para proveer las pólizas de vida y gastos médicos, y aquellas compañías bancarias asignadas para entregar el apoyo económico a eméritos.

Los datos en relación a las compañías son:

- Nombre de la compañía.
- Tipo de compañía.

Permitir la administración de pólizas contratadas

El sistema debe permitir a usuarios administradores el alta y baja de los siguientes datos referentes a las pólizas que son contratadas y asignadas a eméritos:

- Nombre de la póliza.
- Compañía con la que se contrata la póliza.
- Fecha inicio de la vigencia.
- Fecha fin de la vigencia.
- Monto o suma asegurada.

El sistema debe permitir también, la renovación o actualización de pólizas de seguros otorgadas a eméritos.

Generar alertas para notificar información no actualizada o próxima a expirar

Con el objetivo de asegurar que la información sea actualizada constantemente, el sistema debe generar y presentar alertas a usuarios administradores para notificar que existe información de eméritos que no ha sido actualizada recientemente.

Por otra parte, el sistema debe generar alertas para indicar si existen pólizas asignadas a eméritos que están próximas a expirar.

Los datos requeridos para la generación de alertas son:

- Tipo de información a auditar para generar alertas.
- Tiempo establecido para generar alertas.

Permitir la generación de reportes de información de eméritos

El sistema debe permitir a usuarios administradores y de consulta, generar reportes de información referente a eméritos.

Los reportes deben contar con datos agrupados y ordenados en base a:

- Dependencia que propuso las designaciones de eméritos.
- Año de designación de eméritos.
- Género de eméritos.
- Situación de eméritos.
- Año de fallecimiento de eméritos.

Por otra parte, deben existir reportes que permitan visualizar la siguiente información de eméritos:

- Datos para contactar a eméritos, como domicilios o correo electrónico.
- Pólizas de seguros con las que cuentan los eméritos.

Permitir la generación de gráficas de información de eméritos

El sistema debe permitir a usuarios administradores y de consulta, generar gráficas de información referente a eméritos. Los criterios mediante los que se requieren generar las gráficas son:

- Subsistema de la dependencia que propuso las designaciones de eméritos.
- Año de designación de eméritos.

- Género de eméritos.
- Situación de eméritos.
- Año de fallecimiento de eméritos.

Registrar los movimientos realizados por los usuarios en una bitácora

El sistema debe registrar los movimientos de alta y modificación de información en una bitácora con el fin de llevar un control sobre los movimientos realizados por los usuarios, y para determinar que información no es actualizada constantemente.

Los datos que debe registrar la bitácora son:

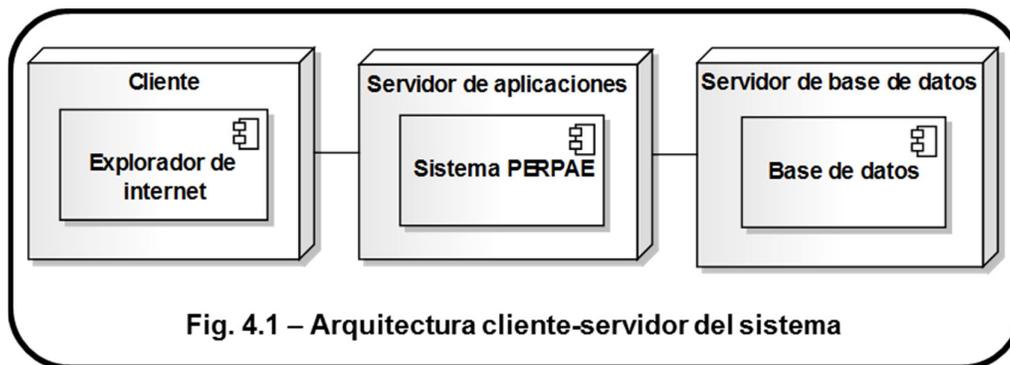
- Usuario que realiza el movimiento.
- Fecha en que se realiza el movimiento.
- Información relacionada al movimiento.

4.1.2 Requerimientos no funcionales

Ser una herramienta web desarrollada en lenguaje *Java*

El sistema debe ser una aplicación web, lo que implica que cuente con una arquitectura cliente servidor y que su acceso sea mediante un navegador de internet.

A continuación se presenta, por medio de un diagrama de implementación, la arquitectura cliente servidor con que debe contar el sistema (Fig. 4.1).



En este diagrama, se presentan tres tipos de nodos o equipos:

- Cliente: Es el equipo de cómputo que cuenta con un navegador de internet y permite a los usuarios finales interactuar con el sistema.
- Servidor de aplicación: Equipo que procesa las peticiones y datos requeridos por el cliente.

- Servidor de base de datos: Equipo que almacena los datos del sistema para ser procesados por el servidor de aplicación.

El servidor de aplicación y el servidor de base de datos deben estar conectados a la red interna de la DGAPA, mientras que el equipo cliente accede al sistema por medio de internet.

Además de ser una aplicación web, el sistema requiere ser desarrollado en lenguaje *Java*, el cual cuenta con las siguientes características:

- Emplea el paradigma orientado a objetos (Ver capítulo II).
- Es multiplataforma, lo que permite implantar programas desarrollados en este lenguaje en distintos tipos de hardware.
- Permite el desarrollo de aplicaciones web y de escritorio.
- Está basado en los lenguajes: C, Cobol y Visual Basic.
- Posee un recolector de basura para gestión de memoria por medio del control del ciclo de vida de los objetos.

Contar con una base de datos en *PostgreSQL*

El sistema debe contar con una base de datos desarrollada en *PostgreSQL* y adaptable a bases de datos ya existentes, lo que implica seguir una serie de normas y estándares propios de la DGAPA, como nombres de estructuras y tipos de datos.

A continuación, se presentan las características de *PostgreSQL*:

- Es un manejador de bases de datos relacional de licencia libre.
- Permite declarar funciones y definir *triggers*.
- Permite la utilización de índices y vistas.
- Permite la gestión de usuarios y permisos.
- Está desarrollado en lenguaje C.
- Puede extenderse con librerías para la encriptación y búsquedas fonéticas.
- Provee manejo de concurrencia.
- Posee algunas extensiones de la orientación objetos, como la herencia.
- Permite la recuperación de transacciones en caso de fallo en su ejecución.
- Soporta una gran variedad de tipos de datos.

Permitir su integración al sistema GeDGAPA

Debido a que el sistema a desarrollar está enfocado en gestionar información del PERPAE, debe estar integrado en el sistema GeDGAPA, el cual concentra sistemas de gestión como:

- PAPIIT: Sistema del Programa de Apoyo a Proyectos de Investigación e Innovación tecnológica.
- PAPIME: Sistema del Programa de Apoyo a Proyectos para la Innovación y Mejoramiento de la Enseñanza.
- INFOCAB: Sistema de la Iniciativa para Fortalecer la Carrera Académica en el Bachillerato de la UNAM
- PASD: Sistema del Programa de Actualización y Superación Docente.

En base a lo anterior, y aunque el sistema a desarrollar maneja procesos independientes de otros sistemas, debe contar con una interfaz gráfica similar y con un modelo de datos adaptable al sistema GeDGAPA.

Proveer un fácil mantenimiento

El sistema debe ser entendible y actualizable de una manera sencilla, lo que implica la generación de documentación que sirva como referencia para su mantenimiento.

En la DGAPA se utilizan estándares como nombres de estructuras de bases de datos, nombres de clases y lógica de programación, que permiten al personal de desarrollo entender el sistema fácilmente, por tal motivo, se debe tomar en cuenta lo anterior para el desarrollo del sistema.

Contar con una interfaz gráfica amigable

El sistema debe contar con una serie de páginas que permitan la realización de procesos de una manera fácil. Además debe contar con elementos gráficos claros y entendibles, y con un flujo de datos sencillo.

Proveer seguridad de la información

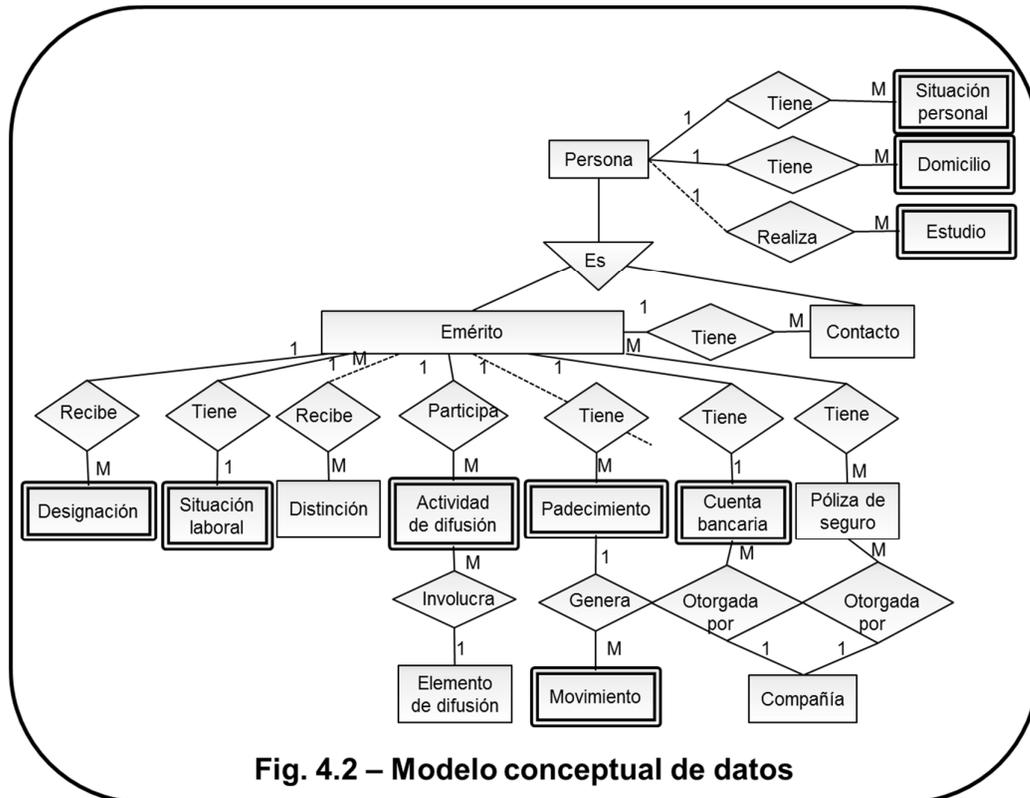
El sistema debe proveer seguridad en la información de las siguientes formas:

- Autenticar a usuarios que requieren ingresar al sistema.
- Proveer seguridad en la barra de direcciones del navegador de internet, para permitir solo a personal autorizado acceder a páginas del sistema.
- El sistema debe validar que en las páginas utilizadas por usuarios administradores y de consulta, solo puedan realizarse las acciones permitidas por cada rol.

4.2 MODELO DE DATOS

4.2.1 Modelo conceptual

A partir del análisis realizado sobre los requerimientos funcionales que involucran la información del PERPAE, se generó el siguiente modelo conceptual que sirvió de base para el diseño de un modelo lógico de datos y posteriormente para la implementación de las estructuras de la base de datos.



4.2.2 Modelo lógico

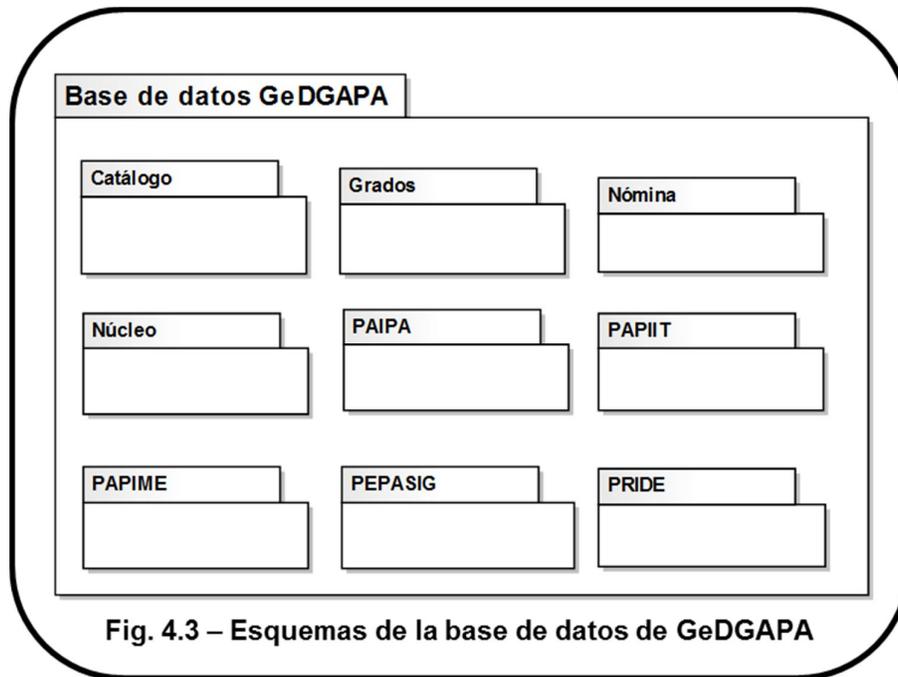
En base al modelo conceptual elaborado, se realizó el modelo lógico de datos, en el cual se diseñó la estructura de las tablas que fueron implementadas en la base de datos del sistema. Para la elaboración de este modelo se siguieron una serie de reglas que son implementadas en la DGAPA para el diseño de bases de datos.

A continuación se presentan los estándares y reglas tomadas en cuenta para el diseño de las estructuras de base de datos del sistema PERPAE:

- Identificar el tipo de estructuras requeridas para almacenar los diferentes tipos de datos.

- El nombre de las tablas cuya función es inserción, consulta y modificación de información, debe comenzar con la letra “t”.
- El nombre de las estructuras que almacenan información de catálogos (estructuras con los diferentes valores que puede tomar un elemento), debe comenzar con la letra “c”.
- El nombre de las llaves primarias de las tablas debe comenzar con “id”, mientras que el nombre de las llaves primarias de los catálogos debe comenzar con “cve”.
- Los catálogos deben tener un campo llamado “estado”, que permita identificar los registros activos, es decir, aquellos que son vigentes.
- Las tablas relacionadas con la información de personas, deben incluir en su nombre la cadena “tipopart”, lo que hace referencia al tipo de participación en el PERPAE.
- En el diseño debe incluirse una descripción general de cada estructura y una descripción de cada uno de sus campos, esto con la finalidad de proveer la documentación necesaria para un adecuado mantenimiento.
- Se debe cuidar la clasificación que se hace entre los campos de una estructura, identificando aquellos que pueden ser nulos y aquellos que no pueden serlo.
- Debe procurarse realizar, además del modelo general de datos, modelos específicos para cada rubro de información. Esto con la finalidad de visualizar de forma más clara las estructuras que serán empleados para la implementación de cada parte del sistema.
- Debe considerarse que existen estructuras que son empleadas para distintos sistemas en la DGAPA y que serán empleadas para el sistema PERPAE. Estas estructuras pertenecen a los esquemas: catálogos, núcleo y nómina.
- Tomar en cuenta el tipo y extensión de dato que corresponde a cada dato.
- Indicar en cada estructura los campos que conforman los índices utilizados para establecer restricciones.
- Establecer un adecuado orden en los campos de cada estructura, dando prioridad a las llaves primarias.

Para la elaboración del modelo lógico se tomó en cuenta la estructura de la base de datos del sistema GeDGAPA, la cual cuenta con distintos esquemas de datos utilizados para dar soporte a diferentes sistemas. A continuación se muestra un diagrama con los esquemas de datos que emplea la base de datos del sistema GeDGAPA:



Para la elaboración del modelo lógico de datos se utilizó el software *ER Studio*, herramienta utilizada para diseñar bases de datos en diferentes plataformas. Esta herramienta permite realizar diseños lógicos y físicos de datos y cuenta con características para realizar actividades de ingeniería inversa. Otra de sus características, es que permite la creación de submodelos de datos y facilita la generación de código de bases de datos.

Para la elaboración del modelo lógico de datos por medio de *ER Studio* se realizó lo siguiente:

- 1) Se creó un nuevo modelo llamado “perpae”.
- 2) Se agregaron al esquema “perpae”, las estructuras ya existentes de los esquemas catálogos, núcleo y nómina que el sistema requiere utilizar.
- 3) Se crearon las estructuras propias del sistema del PERPAE y para cada una, se ingresó una descripción general y una descripción de cada uno de sus campos.
- 4) Se establecieron las relaciones necesarias entre las estructuras.
- 5) Se agregaron a cada estructura los índices requeridos.
- 6) Se realizaron los submodelos de datos, los cuales permiten visualizar las estructuras que pertenecen a los distintos rubros de información, los cuales son:
 - a. Datos personales.
 - b. Situación personal.

- c. Domicilios.
- d. Situación laboral y designaciones.
- e. Distinciones.
- f. Actividades de difusión.
- g. Estudios.
- h. Pólizas de seguros.
- i. Padecimientos.
- j. Catálogos del sistema.
- k. Alertas generadas por el sistema.

El diccionario y los modelos de datos, son presentados en los anexos de este trabajo.

4.3 ELABORACIÓN DE CASOS DE USO

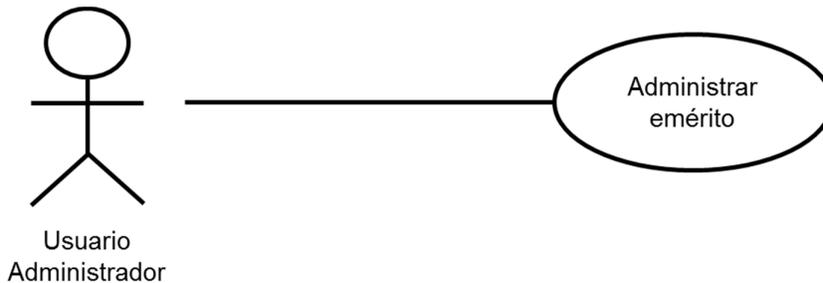
La importancia de realizar los casos de uso de los requerimientos funcionales, radica en que el sistema a desarrollar es un sistema de apoyo a nivel de conocimiento para administración de información, lo que implica que es un sistema orientado a apoyar las tareas de usuarios, y los casos de uso, describen la interacción de usuarios con el sistema, lo cual provee una amplia perspectiva de funcionalidad e interactividad. Por otro lado, los casos de uso proveen apoyo para realizar pruebas del sistema.

En base a lo anterior, se generaron los casos de uso del sistema, para lo que se identificaron los siguientes actores:

- Usuario administrador: Persona capacitada y autorizada para registrar, consultar y dar seguimiento a toda la información que involucra el PERPAE.
- Usuario de consulta: Persona autorizada únicamente para consultar la información estadística y general del PERPAE a través de reportes y gráficas.

A continuación se presentan los diagramas y descripción de algunos de los casos de uso más representativos que corresponden a escenarios derivados de los requerimientos funcionales del sistema.

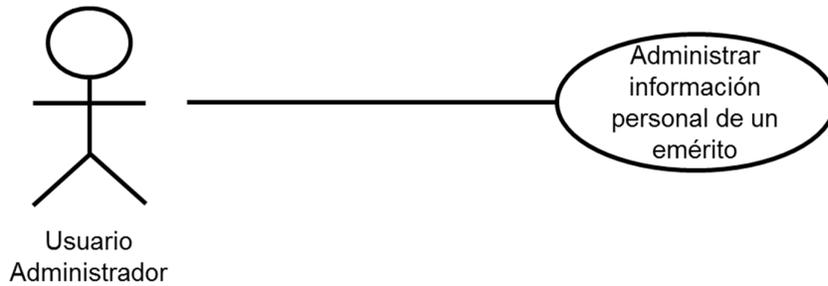
Administrar emérito



CASO DE USO	Administrar emérito
ACTORES	Usuario administrador
DESCRIPCIÓN	Permite al usuario administrador el alta, baja y búsqueda de un emérito en el sistema.
FLUJO BÁSICO "Buscar emérito"	<p>El usuario ingresa a la opción del menú "Eméritos".</p> <p>El sistema presenta un formulario con los siguientes campos: RFC, apellido paterno y número de expediente, que actúan como criterios de búsqueda. Además presenta un botón "Buscar" para cada campo.</p> <p>El usuario realiza la búsqueda de un académico mediante el llenado de un campo y presionando el botón "Buscar" que corresponde.</p> <p>El sistema presenta la lista de los académicos en nómina y la lista de los eméritos del PERPAE, que coinciden con el criterio de búsqueda ingresado.</p>

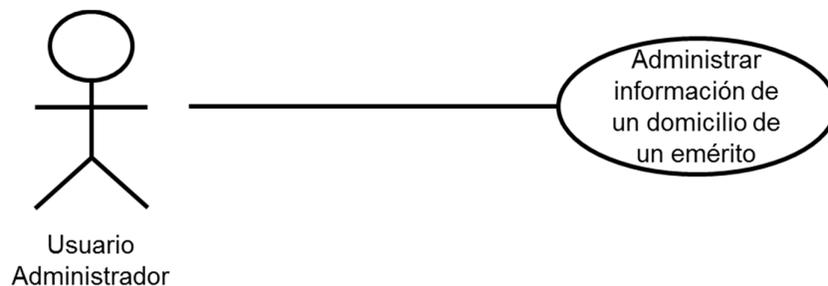
FLUJO EXCEPCIONAL “Error de validación de datos”		El sistema detecta que uno o varios datos no han sido ingresados correctamente y presenta los mensajes de error correspondientes.
PRECONDICIONES		El usuario debe estar autenticado en el sistema y contar con los permisos suficientes.
EXTENSIONES	“Alta de emérito”	El usuario identifica y selecciona por medio de un link, el académico que requiere dar de alta de la lista de académicos en nómina. El sistema presenta la página “Datos personales”. El usuario ingresa los datos del emérito y presiona el botón “Aceptar”. El sistema valida e ingresa los datos en la base de datos del sistema.
	“Presentar menú para administrar datos de emérito”	El usuario selecciona por medio de un link, el emérito requerido de la lista de eméritos del PERPAE. El sistema presenta el menú “Datos de emérito” con las siguientes opciones para administrar información referente al emérito: <ul style="list-style-type: none"> • Datos personales • Situación personal • Domicilios • Contactos • Situación laboral • Distinciones • Difusión • Estudios • Cuenta bancaria • Seguros • Padecimientos El usuario selecciona la opción del menú que requiere. El sistema presenta la página de información correspondiente.
	“Baja de emérito”	El usuario identifica y selecciona mediante una casilla de verificación, al emérito que requiere dar de baja de la lista de eméritos del PERPAE. El usuario presiona el botón “Eliminar”. El sistema presenta una ventana emergente de confirmación. El usuario presiona el botón “Aceptar” de la ventana de confirmación. El sistema da de baja al emérito en el PERPAE.
REGLAS GENERALES		El sistema validará los datos a ingresar y registrará los movimientos de alta en la bitácora del sistema.

Administrar información personal de un emérito



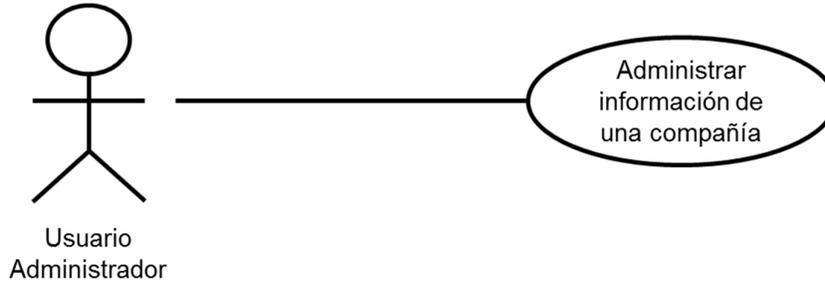
CASO DE USO	Administrar información personal de un emérito
ACTORES	Usuario administrador
DESCRIPCIÓN	Permite al usuario administrador el registro y actualización de datos personales de un emérito en específico.
FLUJO BÁSICO "Administrar información personal"	El usuario ingresa a la página "Datos personales" (Ver caso de uso "Administrar emérito") donde se visualiza un formulario con los datos personales del emérito en cuestión. El usuario ingresa o actualiza los datos personales requeridos y presiona el botón "Aceptar". El sistema valida e ingresa los datos personales del emérito en la base de datos del sistema.
FLUJO EXCEPCIONAL "Error de validación de datos"	El sistema detecta que uno o varios datos no han sido ingresados correctamente y presenta los mensajes de error correspondientes.
PRECONDICIONES	El usuario debe estar autenticado en el sistema y contar con los permisos suficientes.
POSTCONDICIONES	Registro de los datos correspondientes en la base de datos del sistema. Registro de los movimientos realizados en la bitácora del sistema.
REGLAS GENERALES	Las actividades de registro y actualización, serán incluidas en la bitácora del sistema. El sistema validará el formato y tipo de los datos registrados.

Administrar información de un domicilio de un emérito



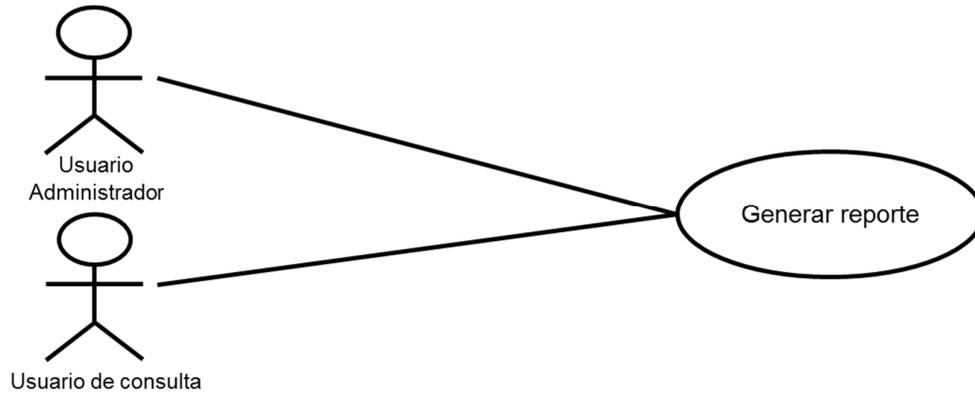
CASO DE USO	Administrar información de un domicilio de un emérito	
ACTORES	Usuario administrador	
DESCRIPCIÓN	Permite al usuario administrador el alta, baja, actualización y consulta de un domicilio de un emérito en específico.	
FLUJO BÁSICO “Alta de domicilio”	<p>El usuario ingresa a la página “Domicilios” donde se visualiza una lista con los domicilios del emérito en cuestión y un formulario para ingresar y visualizar los datos de un domicilio específico.</p> <p>El usuario ingresa los datos de un nuevo domicilio y presiona el botón “Aceptar”.</p> <p>El sistema valida e ingresa los datos en la base de datos del sistema. Posteriormente limpia el formulario y muestra la lista actualizada de domicilios.</p>	
FLUJO EXCEPCIONAL “Error de validación de datos”	El sistema detecta que uno o varios datos no han sido ingresados correctamente y presenta los mensajes de error correspondientes.	
PRECONDICIONES	El usuario debe estar autenticado en el sistema y contar con los permisos suficientes.	
POSTCONDICIONES	<p>Registro de los datos correspondientes en la base de datos del sistema.</p> <p>Registro de los movimientos realizados en la bitácora del sistema.</p>	
EXTENSIONES	“Actualización de domicilio”	<p>El usuario selecciona por medio de un link, el domicilio a actualizar de la lista de domicilios.</p> <p>El sistema muestra los datos correspondientes al domicilio seleccionado.</p> <p>El usuario actualiza los datos del domicilio y presiona el botón “Aceptar”.</p> <p>El sistema actualiza los datos en la base de datos del sistema.</p> <p>El sistema limpia el formulario y muestra la lista actualizada de domicilios.</p>
	“Consulta de domicilio”	<p>El usuario selecciona por medio de un link, el domicilio a consultar de la lista de domicilios.</p> <p>El sistema muestra, en el formulario de la página, los datos correspondientes al domicilio seleccionado.</p> <p>El usuario visualiza los datos del domicilio.</p>
	“Baja de domicilio”	<p>El usuario selecciona por medio de una casilla de verificación, el domicilio a dar de baja de la lista de domicilios, y presiona el botón “Eliminar”.</p> <p>El sistema presenta una ventana emergente de confirmación.</p> <p>El usuario presiona el botón “Aceptar” de la ventana de confirmación.</p> <p>El sistema da de baja el domicilio y muestra la lista actualizada de domicilios.</p>
REGLAS GENERALES	<p>Las actividades de alta y actualización serán incluidas en la bitácora del sistema.</p> <p>El sistema validará el formato y tipo de los datos registrados.</p>	

Administrar información de una compañía



CASO DE USO	Administrar información de una compañía.
ACTORES	Usuario administrador
DESCRIPCIÓN	Permite al usuario administrador el alta y baja de una compañía. Además, de permitir visualizar el catálogo de compañías del sistema.
FLUJO BÁSICO "Alta de compañía"	El usuario ingresa a la página "Compañías", seleccionando la opción del menú "Catálogos" submenú "Compañías", donde se visualiza una lista con las compañías existentes y un formulario para ingresar los datos de una nueva compañía. El usuario ingresa los datos correspondientes a la compañía a agregar y presiona el botón "Aceptar". El sistema valida e ingresa los datos en la base de datos del sistema. Posteriormente, muestra la lista actualizada de compañías, y limpia el formulario de la página.
FLUJO EXCEPCIONAL "Error de validación de datos"	El sistema detecta que uno o varios datos no han sido ingresados correctamente y presenta los mensajes de error correspondientes.
PRECONDICIONES	El usuario debe estar autenticado en el sistema y contar con los permisos suficientes.
POSTCONDICIONES	Registro de los datos correspondientes en la base de datos del sistema.
EXTENSIÓN "Baja de compañía"	El usuario selecciona por medio de una casilla de verificación, la compañía a dar de baja de la lista de compañías, y presiona el botón "Eliminar". El sistema presenta una ventana emergente de confirmación. El usuario presiona el botón "Aceptar" de la ventana de confirmación. El sistema da de baja la compañía y posteriormente muestra la lista actualizada de compañías.
REGLAS GENERALES	El sistema validará el formato y tipo de los datos registrados. El sistema no permitirá la baja de una compañía asociada a eméritos.

Generar reporte



CASO DE USO	Generar reporte
ACTORES	Usuario administrador y usuario de consulta
DESCRIPCIÓN	Permite a usuarios administradores y a usuarios de consulta generar un reporte de información general de eméritos.
FLUJO BÁSICO "Generar reporte"	El usuario ingresa a la página "Reportes" seleccionando la opción del menú "Reportes", donde se visualiza la lista de los posibles reportes a generar. El usuario selecciona el reporte requerido. El sistema genera y presenta el reporte en pantalla.
PRECONDICIONES	El usuario debe estar autenticado en el sistema y contar con los permisos suficientes.
POSTCONDICIONES	Generación de un reporte.

4.4 DISEÑO DE LA INTERFAZ GRÁFICA

El diseño de la interfaz gráfica es una actividad importante en el diseño del sistema, ya que muestra la parte que será visualizada por el usuario y sirve como base para la programación de los módulos del sistema.

De acuerdo con la información relacionada al PERPAE y analizada anteriormente, se establecieron las siguientes interfaces gráficas para ser agregadas al sistema:

Página	Funciones
Inicio	Bienvenida y visualización de alertas en general.
Eméritos	Búsqueda, alta y baja de eméritos.
Datos personales	Administración de la información personal de eméritos.
Situación personal	Administración de la situación personal de eméritos.
Domicilios	Administración de los domicilios de eméritos.
Contactos	Administración de los contactos de eméritos.
Situación laboral	Administración de la información laboral y designaciones de eméritos.
Distinciones	Administración de las distinciones otorgadas a eméritos.
Difusión	Administración de las actividades o elementos de difusión en que participan eméritos.
Estudios	Administración de los estudios realizados por eméritos.
Cuenta bancaria	Administración de la cuenta bancaria de eméritos.
Pólizas	Administración de las pólizas de seguros de eméritos.
Póliza de vida	Administración de los seguros de vida de eméritos.
Póliza de gastos médicos	Administración de los seguros de gastos médicos de eméritos.
Padecimientos	Administración de los padecimientos de eméritos.
Movimientos por padecimiento	Administración de los movimientos por padecimientos de eméritos.
Actualización de pólizas	Actualización de pólizas de eméritos.
Catálogo de pólizas	Administración del catálogo de pólizas.
Catálogo de compañías	Administración del catálogo de compañías.
Catálogo de elementos de difusión	Administración del catálogo de elementos de difusión.
Catálogos de distinciones	Administración del catálogo de distinciones.
Pólizas contratadas	Administración de las pólizas contratadas con una compañía.
Reportes	Generación de reportes de información de eméritos.
Gráficas	Generación de gráficas de información de eméritos.
Alertas	Visualización de las alertas a detalle.

Tabla 4.1 – Páginas del sistema PERPAE.

Además de estas páginas, se tomaron en cuenta las siguientes páginas del sistema GeDGAPA, por medio de las cuales, se podrá acceder al sistema PERPAE:

Página	Funciones
Registro	Autenticación e ingreso al sistema GeDGAPA.
Menú principal	Visualización de los sistemas de GeDGAPA a los que se puede acceder.

Tabla 4.2 – Páginas del sistema GeDGAPA.

Para el diseño de las interfaces, se comenzó por realizar prototipos en papel debido a la facilidad y rapidez que implica, y como apoyo para visualizar el número y tipo de elementos requeridos en cada página. Además, el flujo de páginas pudo ser visualizado de una forma sencilla y quedando establecido de la siguiente manera:

Para usuarios administradores:

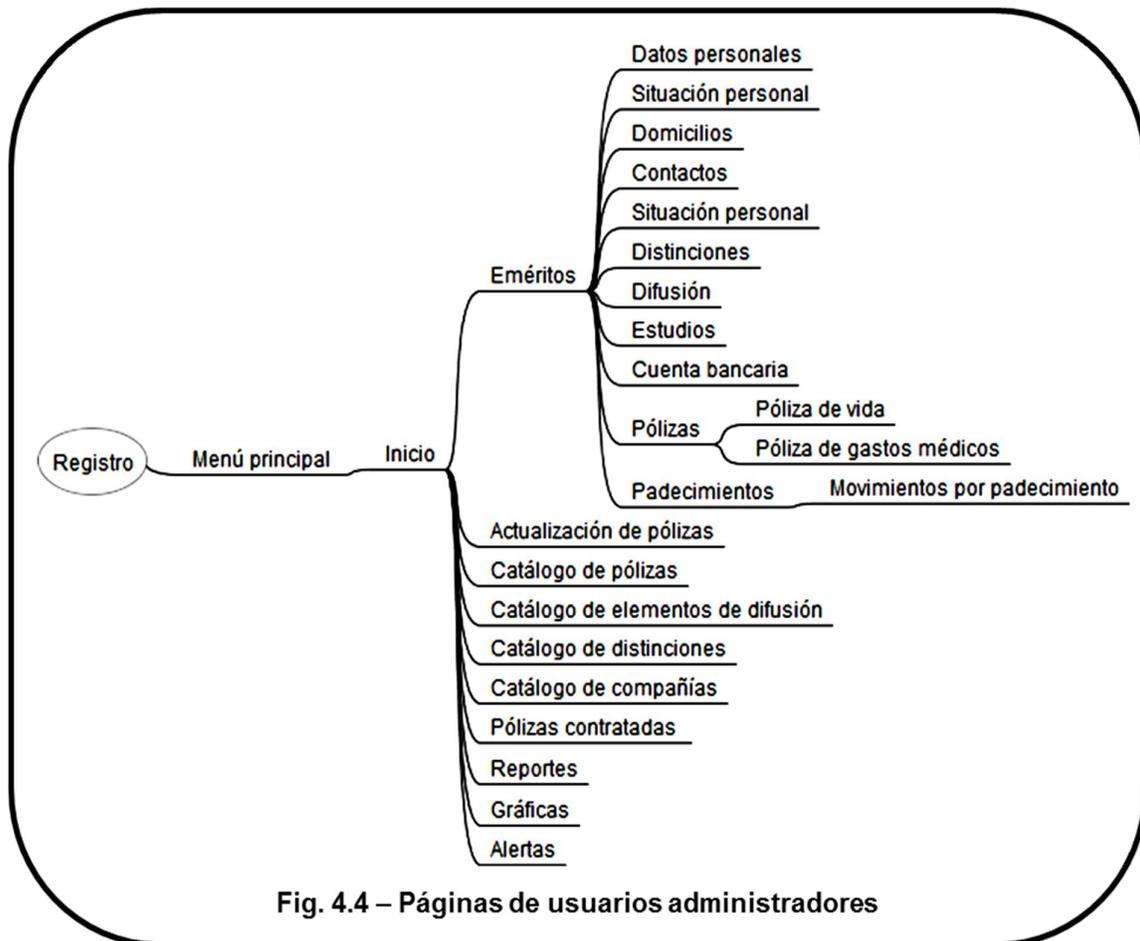


Fig. 4.4 – Páginas de usuarios administradores

Para usuarios de consulta:

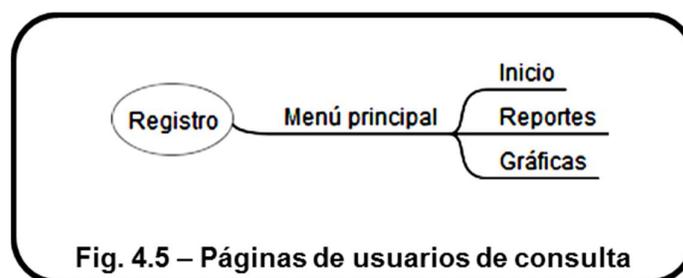


Fig. 4.5 – Páginas de usuarios de consulta

En base al diseño de las páginas realizado en papel, se procedió a realizar el diseño mediante código web. Lo anterior se hizo con la finalidad de que el diseño realizado sirviera como base para la implementación del sistema y para evitar duplicar trabajo al hacerlo con una herramienta de diseño y posteriormente con una herramienta de desarrollo.

Para el diseño de las páginas se tomaron en cuenta las siguientes consideraciones:

- Para proveer coherencia entre las páginas y la base de datos, cada elemento de la interfaz gráfica debe ser compatible con el tipo de dato al que está relacionado, para esto, se estableció la siguiente relación entre elementos gráficos y tipos de datos:

Tipo de elemento gráfico	Tipos de datos relacionados
Campo de texto	Cadenas alfanuméricas
Campo de fecha	Fechas
Área de texto	Textos o cadenas grandes
Casilla de verificación	Lógicos o con dos posibles valores
Lista desplegable	Con varios valores posibles
Etiqueta	Datos de lectura o calculados
Cuadrícula	Tablas o matrices de información

Tabla 4.3 – Elementos gráficos y tipos de datos.

- Las páginas deben permitir a los usuarios realizar acciones como el alta, baja, consulta y actualización de información, para esto, se emplearon los siguientes elementos:

Tipo de elemento gráfico	Acciones que permite realizar
Botón	Ingresar información Actualizar información Eliminar información Limpiar información
Vínculo	Consultar información Acceder a una nueva página

Tabla 4.4 – Elementos gráficos y acciones.

- Para evitar errores durante el registro de información, se optó por manejar listas desplegables en lugar de cuadros de texto para algunos datos.
- Para proveer amigabilidad, se incluyeron en las páginas mensajes para indicar a los usuarios lo siguiente:
 - Si el usuario se encuentra realizando actividades de consulta o actualización de información.
 - El formato requerido de los datos para su captura.
 - La forma de inserción de datos.
 - Los datos cuyo valor es requerido para el registro de información.
- Presentar la información agrupada y de forma clara.
- Incluir títulos claros y referentes al uso de cada página.
- Incluir en las páginas de administración de información de eméritos, un encabezado que permita visualizar la información más representativa del emérito en cuestión, tal como nombre, RFC, número de expediente, edad, género y situación en general.
- Con el fin de facilitar el aprendizaje y uso del sistema, los elementos de las páginas fueron adaptados siguiendo el mismo patrón. También, se siguió el mismo patrón para permitir la inserción, eliminación, consulta y actualización de información en cada página.
- El menú del sistema fue ordenado de manera lógica y ajustado a un tamaño adecuado.
- Se presentó un encabezado general que indica el usuario y tipo de usuario que ha ingresado al sistema.
- Se utilizaron solamente los colores e imágenes necesarios en cada página para evitar la sobrecarga de elementos y adecuar el diseño para su integración al sistema GeDGAPA.

El diseño de las interfaces gráficas tomando en cuenta las consideraciones anteriores se presentan en anexos de este trabajo.

4.5 DISEÑO DE REPORTE

Para el diseño de los reportes, se identificaron los elementos o partes que debían conformarlos con la finalidad de organizarlos y ajustarlos de acuerdo a las características de una hoja tamaño carta, que es el medio físico donde deben ser generados.

Las partes que se identificaron para conformar los reportes son las siguientes:

- Encabezado: Parte presentada en la parte superior del reporte que contiene una imagen de la UNAM e información general del reporte.
- Detalle del reporte: Parte que contiene los datos que conforman el reporte, tales como información de eméritos.
- Resumen: Parte que se visualiza en la última página del reporte y contiene un resumen de la información presentada.
- Pie de página: Se visualiza en la parte inferior de la página y contiene el número de página y la fecha de generación del reporte.

A continuación se muestra el diseño propuesto de los reportes:

	UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO DIRECCIÓN GENERAL DE ASUNTOS DEL PERSONAL ACADÉMICO PROGRAMA DE ESTÍMULOS Y RECONOCIMIENTOS AL PERSONAL ACADÉMICO EMÉRITO TÍTULO DEL REPORTE
	<hr/> <p>LISTADO DE INFORMACIÓN</p> <p>REGISTRO DE INFORMACIÓN DEL PERPAE 1</p> <p>REGISTRO DE INFORMACIÓN DEL PERPAE 2</p> <p>REGISTRO DE INFORMACIÓN DEL PERPAE 3</p> <p>REGISTRO DE INFORMACIÓN DEL PERPAE 4</p> <p>REGISTRO DE INFORMACIÓN DEL PERPAE 5</p> <p>RESUMEN DE INFORMACIÓN</p>
PÁGINA # DE #	FECHA DE GENERACIÓN DEL REPORTE

Fig. 4.6 – Diseño de reportes

4.6 DISEÑO DE GRÁFICAS

Para el diseño de las gráficas, se tomó en cuenta que los datos a representar son organizados en pocos grupos. Debido a esto, y en base al uso que los usuarios requieren darle a las gráficas, se seleccionó el tipo de gráficas de barras para representar la información.

Debido a que los usuarios también requieren visualizar la información de las gráficas en tablas para manejar los datos de una forma más sencilla, se diseñaron reportes para presentar de manera conjunta las gráficas y las tablas de información asociadas. A continuación se muestra el diseño de los reportes de gráficas propuesto:

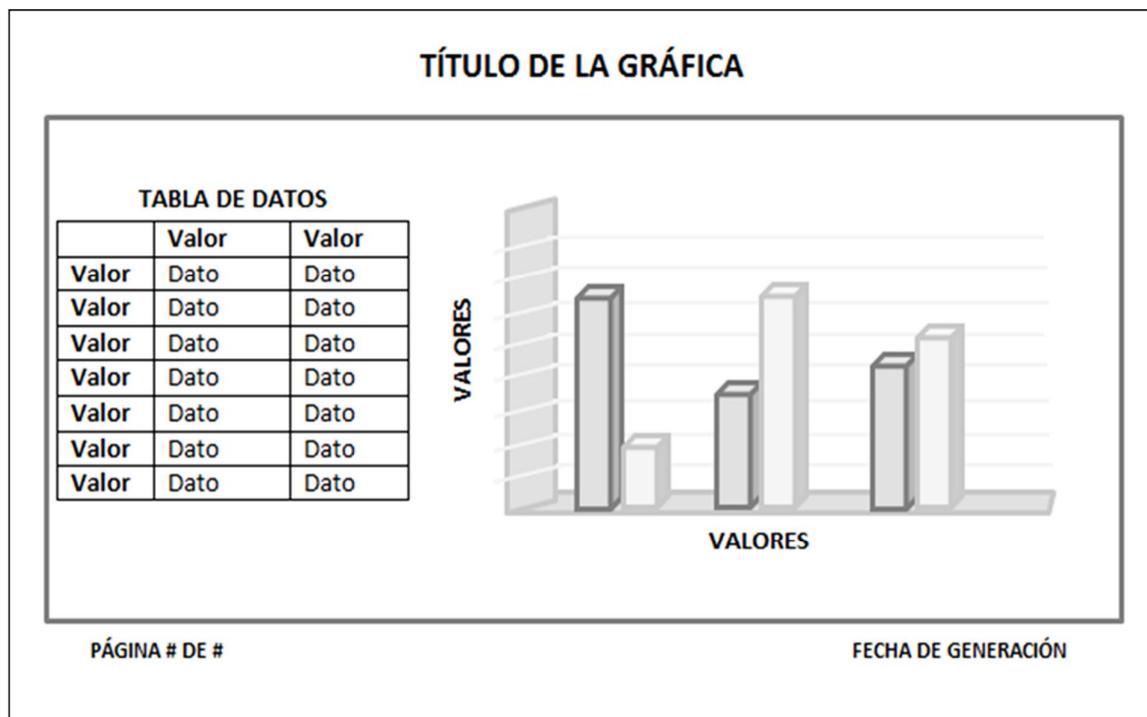


Fig. 4.7 – Diseño de gráficas

CAPÍTULO V

DESARROLLO

En este capítulo se presentan las actividades realizadas para el desarrollo del sistema de información del PERPAE, tales como implementación de la base de datos y codificación. Estas actividades se sustentaron en el análisis y diseño previamente realizado y dieron como resultado la generación del sistema requerido.

5.1 TECNOLOGÍA Y LOGÍSTICA

Para las actividades de desarrollo del sistema se utilizaron herramientas tecnológicas que dieron soporte al sistema y facilitaron su implementación. A continuación se describen de manera general y se justifica su uso.

5.1.1 Software

Eclipse

Para el desarrollo del sistema se requirió una herramienta que permitiera visualizar y generar de forma sencilla el código de cada elemento involucrado, además debía permitir el desarrollo de aplicaciones web en lenguaje *Java*.

En base en lo anterior, fue utilizado *Eclipse* como IDE (*Integrated Development Environment* o entorno de desarrollo integrado) para el desarrollo del sistema.

Eclipse es una herramienta multiplataforma para desarrollo de aplicaciones de cliente enriquecido, para lo que emplea módulos o *plug-ins* para proporcionar dicha funcionalidad. Ofrece soporte para desarrollo en *Java*, además permite extender su uso a otros lenguajes de programación como *C*, *C++* y *Python*. Dispone de un editor de texto con resaltado de sintaxis, compilación en tiempo real, control de versiones y asistentes para la creación de proyectos. También puede integrar el control de versiones *Subversion* y la herramienta *Hibernate*.

Tapestry

Debido a que el uso de un componente web facilita la implementación y provee mayor funcionalidad a un sistema, fue utilizado *Tapestry* como *framework* para el desarrollo del sistema del PERPAE.

Tapestry permite crear aplicaciones web de forma dinámica, robusta y escalable en lenguaje *Java*. Su desarrollo se basa en términos de objetos, métodos, propiedades y eventos.

Permite dividir una aplicación en un conjunto de páginas que se constituyen de componentes, los cuales pueden generar diversos eventos ante los que la página reacciona.

Tapestry provee una alta productividad y puede trabajar en conjunto con otros *frameworks* como *Spring* e *Hibernate*.

Dentro de sus ventajas, se encuentran que es ligero, escalable y facilita la separación de código e interfaz. Además, la gran variedad de componentes propios de *Tapestry* pueden ser embebidos dentro de código HTML. Algunas de sus desventajas son la curva de aprendizaje y la falta de documentación.

Maven

La herramienta *Maven* fue empleada debido a que facilita el desarrollo de proyectos con *Teastry* y provee soporte para el manejo de distintas tecnologías en conjunto.

Maven facilita la gestión de proyectos, provee soporte para aplicaciones web, y permite el desarrollo en lenguaje *Java*. Sus objetivos principales son la compilación de código y el empaquetado.

La filosofía de *Maven* se basa en la construcción de un proyecto con una estructura ya definida utilizando un POM (*Project Object Model* o Modelo de Objetos de Proyecto) para describir el software, sus dependencias, componentes externos y el orden de construcción de sus elementos por medio de XML.

Hibernate

Para el desarrollo del sistema se requirió una herramienta que permitiera el acceso y manejo de la base de datos de una forma sencilla y que fuera compatible con *Java*. Por esta razón se utilizó *Hibernate*.

Hibernate es una herramienta de mapeo para la plataforma *Java*, empleada para mapear los atributos de una base de datos relacional a un modelo de objetos de una aplicación por medio de archivos declarativos XML. Esto facilita la manipulación de bases de datos, ya que se utilizan las características de la programación orientada a objetos.

Hibernate funciona asociando cada tabla de la base de datos a un POJO (*Plain Old Java Object*) por medio de ficheros *.hbm.xml*. Los POJO's son clases en lenguaje *Java* que no dependen de ningún *framework*.

Subversion

Una cuestión importante para el desarrollo del sistema, es el uso de una herramienta que facilite controlar y respaldar las versiones del proyecto. Para esto fue utilizado *Subversion*, que es un sistema de versiones que permite acceder al repositorio de un proyecto a través de una red y permite versionar operaciones de copiado, borrado y renombrado de elementos

Subversion es un software de licencia libre y puede utilizarse mediante el IDE *Eclipse*.

Apache

Para el desarrollo del sistema se requirió una herramienta que soportara aplicaciones web, por lo cual se utilizó *Apache*, que es un servidor http empleado para el desarrollo de aplicaciones web, y cuya principal función es el envío de páginas web a la *World Wide Web* (WWW). Este servidor fue seleccionado debido a que ofrece grandes ventajas como ser modular, de código abierto, multiplataforma, extensible y robusto, además de ser empleado en la DGAPA para dar soporte a otros sistemas.

PgAdmin

Debido a que durante el desarrollo del sistema fueron realizadas actividades sobre la base de datos del sistema, era requerida una herramienta de administración de bases de datos.

Para lo anterior se utilizó *PgAdmin*, herramienta que fue seleccionada debido a que gestiona principalmente bases de datos de *PostgreSQL* e incluye una amplia variedad de funcionalidades. Además *PgAdmin* es una herramienta de uso libre compatible con distintas plataformas.

JasperReports

Debido a que uno de los requerimientos del sistema es la generación de reportes y gráficas, era necesaria una herramienta que facilitara esta tarea y que fuera compatible con *Java*. Para esto se utilizó *JasperReports*, que es un software de licencia libre empleado para la creación de informes, escrito en lenguaje *Java* y compatible con aplicaciones escritas en dicho lenguaje.

JasperReports provee distintas funciones para trabajar con reportes y gráficas, como la edición, impresión y exportación a distintos formatos. Para esto utiliza una interfaz gráfica amigable llamada *iReport*.

5.1.2 Hardware

Equipo de desarrollo

Para el desarrollo del sistema fue utilizado un equipo de cómputo asignado por la DGAPA con las siguientes características:

- Sistema operativo Ubuntu.
- Procesador Intel Core 2.
- 2GB de memoria RAM.
- 160GB de memoria en disco duro.
- Conexión a la red interna de la DGAPA y a Internet.
- Navegador de internet.

Servidores

Para dar soporte a la base de datos del sistema, se utilizó un servidor de prueba que aloja bases de datos en *PostgreSQL*. Este servidor cuenta con las siguientes características:

- 2 procesadores Intel(R) Xenon(TM)
- CPU E5620 con 2.40 GHz
- 16 GB RAM
- 260 GB de disco duro con conexión de tipo IDE (*Integrated Drive Electronics* o dispositivo electrónico integrado)
- 6 tarjetas de red
- Sistema operativo Ubuntu 10.04 Lucid

5.2 IMPLEMENTACIÓN DE LA BASE DE DATOS

La base de datos es una parte fundamental del sistema a desarrollar, debido a que da soporte para la administración de información. Su implementación se basó en la generación de un modelo físico, que permitió describirla desde una perspectiva del gestor de base de datos. Debido a esto, fue necesario considerar las características de *PostgreSQL 8.2.4*, que es el sistema gestor de base de datos que se empleó en el sistema del PERPAE.

Tomando en cuenta las características de *PostgreSQL* y en base al modelo lógico de datos realizado con anterioridad, se generó el modelo físico de datos, que sirvió para la implementación de las estructuras de la base de datos.

5.2.1 Modelo físico de datos

La elaboración del modelo físico de datos facilitó la generación del código requerido para la creación de las estructuras de la base de datos del sistema. Para su elaboración, se empleó la herramienta *ER Studio*, misma que fue utilizada para la elaboración del modelo lógico descrito en el capítulo anterior.

ER Studio permitió generar de forma sencilla el modelo físico de datos a partir del modelo lógico realizado. Para esto, se seleccionó el modelo lógico y se seleccionó la opción “*Generate physical model*”, con lo cual *ER Studio* presentó una serie de ventanas que permitieron seleccionar e ingresar las características con que contaría el modelo.

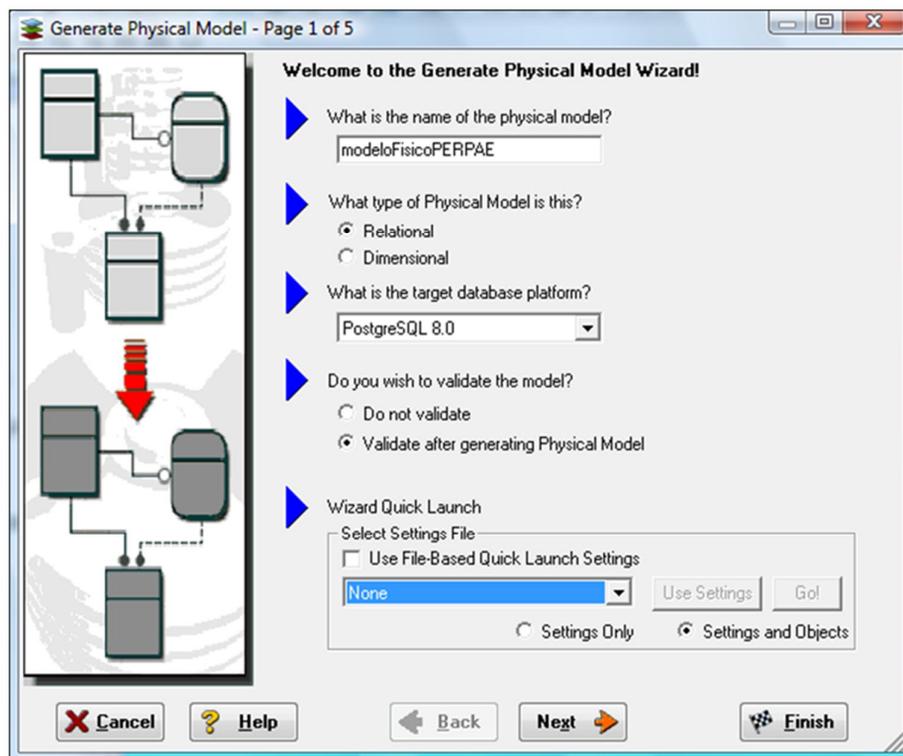


Fig. 5.1 – Interfaz de ER Studio

El modelo físico generado, contó con los tipos de datos propios de *PostgreSQL*, además incluyó los índices y restricciones de cada estructura. Este modelo también permitió visualizar el código DDL de cada estructura.

5.2.2 Generación de código e implementación

Para la implementación de las estructuras de la base de datos, se requirió del código DDL, el cual incluye las sentencias para la creación de dichas estructuras.

El código DDL fue generado a partir del modelo físico con ayuda de la herramienta *ER Studio*. Para esto se seleccionó la opción “*Generate Database*” del modelo físico anteriormente creado, con lo que se visualizó un asistente que permitió indicar las características que se incluyeron en la generación del código.

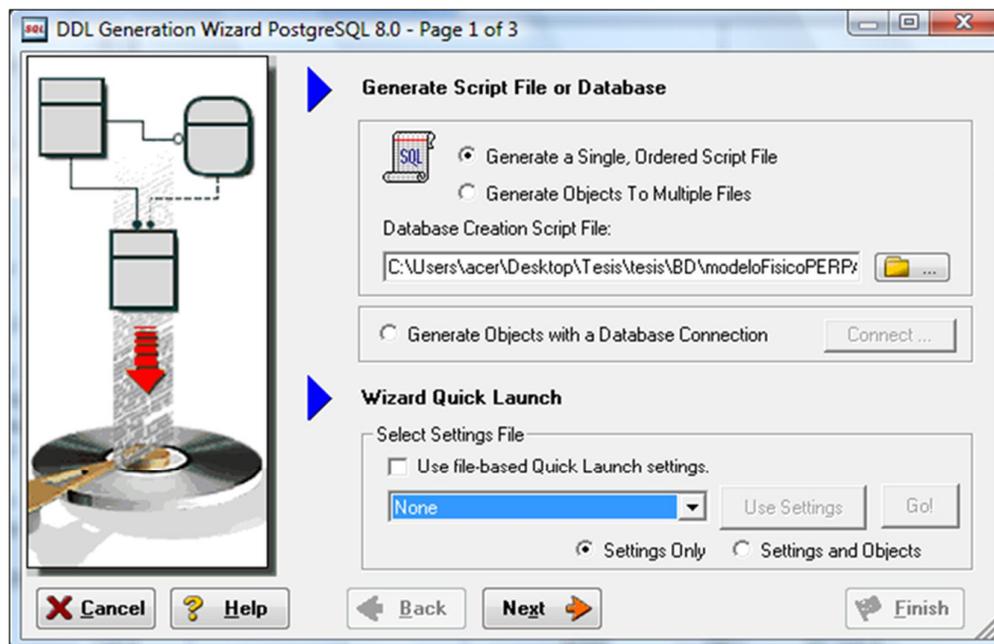


Fig. 5.2 – Interfaz de ER Studio

Para la generación del código DDL, se tomó en cuenta lo siguiente:

- Solo se incluyeron las estructuras propias del PERPE, ya que las estructuras de los esquemas de catálogos, nómina y núcleo, ya estaban implementadas en la base de datos.
- Se incluyó el código el DDL para las restricciones de estructuras, como llaves foráneas e índices.
- Se incluyeron las descripciones de estructuras y campos de las mismas.

El código DDL generado fue almacenado en un archivo de texto que fue revisado antes de ser ejecutado para garantizar que fuera correcto. Para esto se tomaron las siguientes precauciones:

- Se cuidó que el código empleara únicamente letras minúsculas.
- Se cuidó que el tipo de dato de cada campo fuera el correcto.
- Se cuidó que el tipo de dato de las llaves primarias fuera de tipo serial, lo cual facilitó la inserción y modificación de datos ya que este tipo de dato es autoincremental y evita el cálculo del valor de la llave primaria.
- El nombre de las llaves primarias de las estructuras siguió el formato: pk_nombreestructura.
- El nombre de las llaves foráneas siguió el formato: fk_nombreestructura1_nombreestructura2
- El nombre de los índices siguió el formato: indx_nombreestructura.

Una vez revisado el código, se empleó la herramienta *PgAdmin*, mediante la cual se ingresó y ejecutó el código DDL.

Con la ejecución del código, se crearon las estructuras que dieron soporte al sistema. Estas estructuras se crearon en un esquema llamado “perpae”, en una base de datos general de un servidor de pruebas.

Una vez creadas las estructuras del sistema, se otorgaron permisos para su gestión por medio de la herramienta *PgAdmin*.

Posteriormente se procedió a ingresar los datos correspondientes a las estructuras de catálogos del sistema perpae por medio de sentencias de inserción.

5.3 GENERACIÓN Y CONFIGURACIÓN DEL PROYECTO

5.3.1 Generación del proyecto

La generación de un proyecto con ayuda de *Maven* sirve como base para la codificación del sistema, ya que permite generar un esqueleto con una estructura que permite organizar los diferentes tipos de archivos que conforman el proyecto.

Para la generación del esqueleto del proyecto se utilizó una terminal, en donde se ejecutó el siguiente comando:

```
mvn archetype:generate -DarchetypeCatalog= http://tapestry.apache.org.
```

Durante la generación del proyecto se ingresaron mediante la terminal, las siguientes características:

- Versión de Tapestry a utilizar: 5.3.3
- Nombre del proyecto: perpae
- Paquete que contiene el proyecto: mx.unam.dgapa

Como resultado de lo anterior, se generó el esqueleto del proyecto con una estructura base que agrupa los archivos y elementos de la siguiente forma:

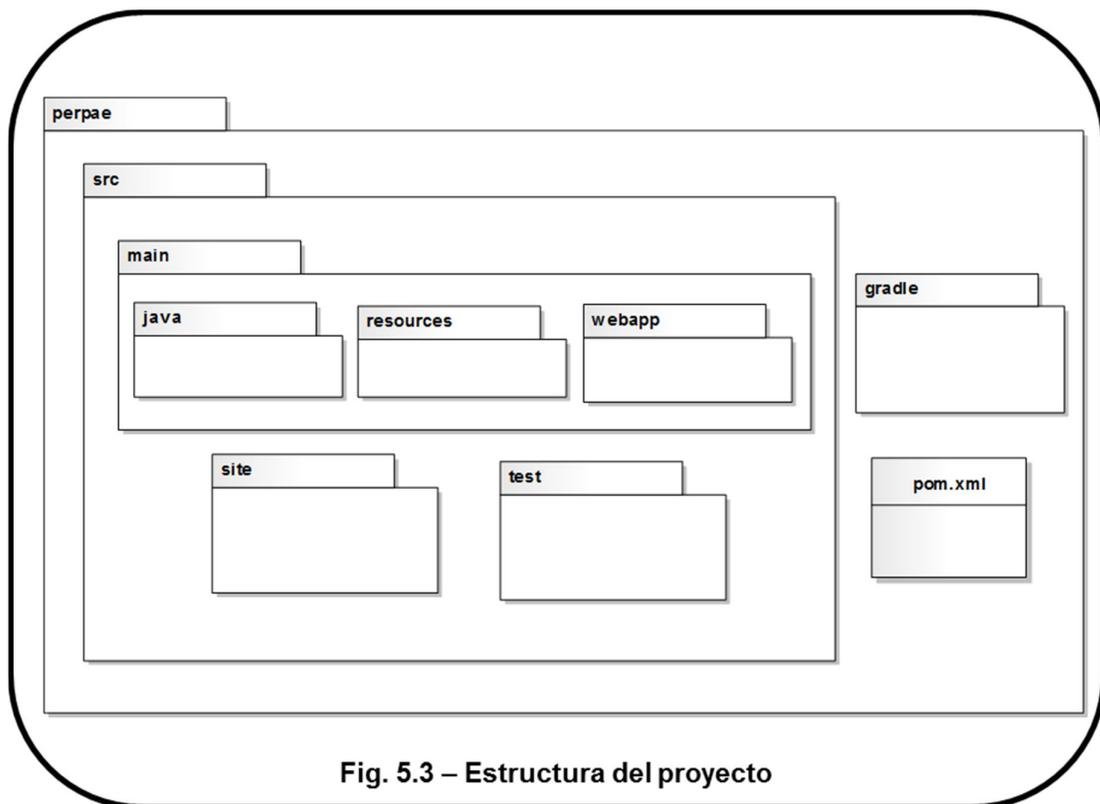


Fig. 5.3 – Estructura del proyecto

El esqueleto del proyecto generado con *Tapestry* cuenta con una estructura que permite separar los diferentes archivos con que se componen las páginas. Estos son de tres tipos:

- Plantillas: Contienen el código propio de los elementos gráficos de la página y poseen la extensión `.tml`. Son esencialmente código HTML con características para vincular los elementos gráficos con una clase *Java* asociada y hacer uso de los componentes de *Tapestry*.
- Clase *Java*. Son los archivos asociados a las plantillas de la página, las cuales contienen las propiedades y métodos de las mismas. El nombre de la clase debe coincidir con el nombre de la plantilla asociada, por ejemplo, una página con una plantilla llamada `Index.tml` debe tener asociada una clase llamada `Index.java`.
- Archivo de propiedades: Son los archivos para manejar las características de los componentes de la página, tales como nombre de etiquetas o mensajes de validación. Poseen la extensión `.properties`.

Por otra parte, el proyecto generado cuenta con las siguientes características:

- Contiene componentes *gradle* para definir el ciclo de vida del proyecto.
- La ubicación `src\main\java\mx\unam\dgapa\perpae` contiene directorios para almacenar los archivos `.java` asociados a los componentes de páginas de la aplicación.
- La ubicación `src\main\resources\mx\unam\dgapa\perpae` contiene directorios para almacenar los archivos `.tml` asociados a los componentes web del sistema y archivos `.properties` asociados a las páginas del sistema.
- Cuenta con un archivo `pom.xml` con información sobre el proyecto, tal como detalles de configuración, *plug-ins* utilizados y dependencias y versión del proyecto.

5.3.2 Configuración del proyecto

Una vez generado el esqueleto del proyecto, se agregó al ambiente de *Eclipse*, mediante el cual se implementaron las páginas y elementos del sistema.

Para comenzar a trabajar sobre el proyecto, se realizaron las siguientes adecuaciones:

- En el archivo `pom.xml` se agregó el código para que el proyecto integrara componentes de *Hibernate*, *PostgreSQL*, *Java*, *Maven* y *JasperReports*.
- Se agregó una carpeta a la ubicación `src\main\` llamada "config", en donde se agregó un documento XML llamado "jetty-env", el cual es requerido para establecer la conexión con la base de datos. En este archivo se ingresó el código con los parámetros de conexión a la base de datos tales como:

- Nombre del usuario de la base de datos.
 - Contraseña.
 - Nombre de la base de datos.
 - IP del servidor que almacena la base de datos.
 - Número del puerto para el acceso.
- En la ubicación `\src\main\java\mx\unam\dgapa\perpae\services` se agregó una carpeta llamada “properties”, con una carpeta llamada “texto” para almacenar archivos .java para el manejo de constantes del sistema, y una carpeta llamada “xml” para almacenar archivos .java para manejar la base de datos.
 - En la ubicación `\src\main\webapp` se agregó:
 - Una carpeta llamada “admin” para manejar los archivos .tml de las páginas.
 - Una carpeta “borde” para almacenar imágenes y archivos CSS.
 - Una carpeta llamada “configuración”, con una carpeta “properties” para almacenar archivos de propiedades generales del sistema y una carpeta “*queries*” para almacenar los archivos .xml con los *queries* que requiere ejecutar el sistema.

5.3.3 Generación del mapeo de la base de datos

El mapeo de la base de datos mediante *Hibernate* facilitó la manipulación de datos durante la codificación del sistema ya que permitió simular las estructuras de la base de datos a través de clases que pudieron ser manipuladas por medio de *Java*.

Para diversos sistemas de la DGAPA, se emplea un mapeo general que contiene diversos esquemas de datos, por esta razón, se requirió generar el mapeo correspondiente al esquema de datos del PERPAE e incluirlo en el mapeo general.

Para la generación del mapeo de las estructuras del PERPAE, se crearon los siguientes tipos de archivos:

- Clases *Java*: Permiten manejar las estructuras de la base de datos mediante la programación orientada a objetos. Contienen un constructor sin parámetros y una serie de atributos, a los que se accede por medio de métodos `set()` y `get()`. Cada una de las clases generadas representa una estructura de la base de datos, así que por cada estructura, se creó una clase *Java*.
- Archivos de mapeo: Son archivos que contienen el mapeo de datos, el cual representa la relación que existe entre las clases *Java* y cada una de las estructuras de la base de datos. Estos archivos son ficheros XML y tienen

extensión `.hbm.xml` que hace referencia a archivos propios de *Hibernate*. Algunas de las etiquetas utilizadas en este tipo de archivos son:

- Class: Indica el nombre de la clase *Java* y la estructura relacionada.
- Id: Indica el nombre y tipo de dato de la llave primaria de una estructura.
- Property: Indica el nombre y tipo de los atributos de una estructura.
- Set: Indica relaciones entre estructuras.

Una vez creado el mapeo de las estructuras del PERPAE, se agregó al mapeo general llamado “gedgapa-hibernate”, el cual cuenta con una estructura que agrupa los archivos `.java` de las clases asociadas y los archivos de mapeo.

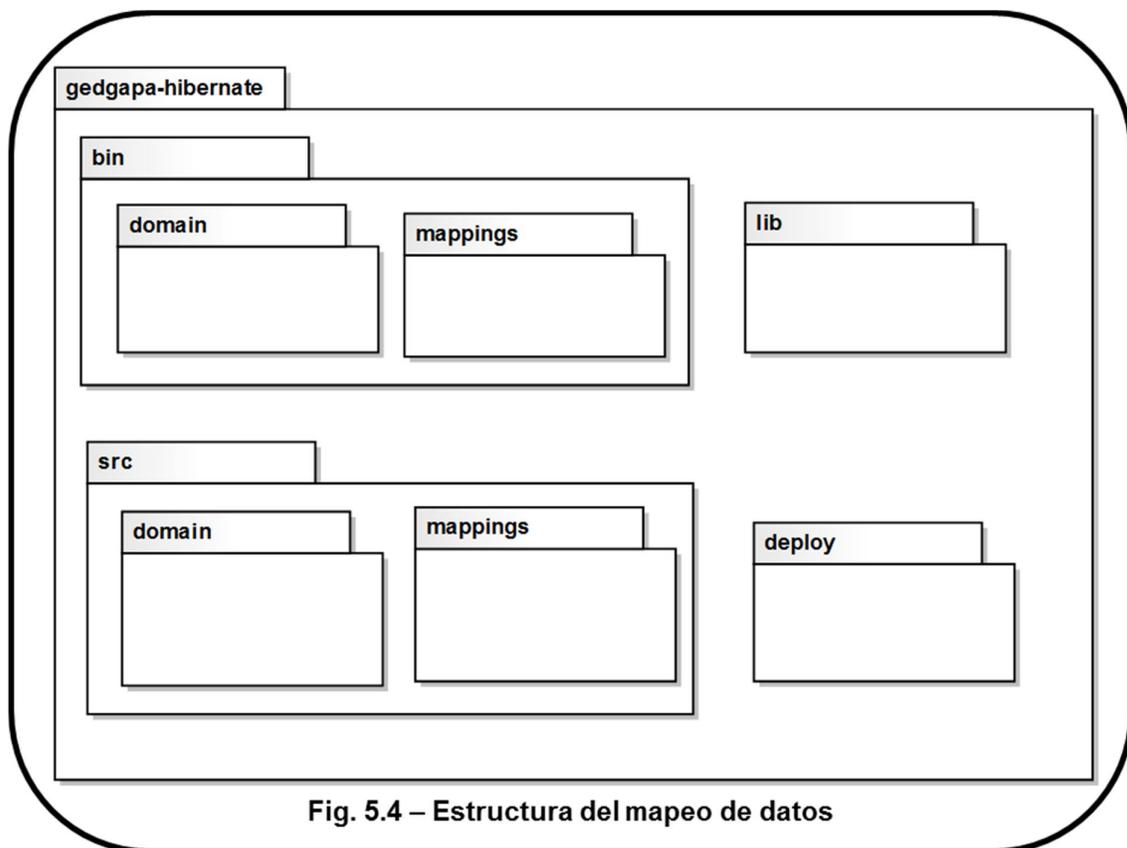


Fig. 5.4 – Estructura del mapeo de datos

Algunas de las características del mapeo presentado en la imagen anterior son:

- Los directorios “domain” y “mappings” del directorio “bin” contienen los archivos `.class`, que son los archivos de las clases *Java* ya compiladas.
- El directorio “lib” contiene los archivos `.jar` propios de *Hibernate*.
- Los directorios “domain” y “mappings” del directorio “src” contienen los archivos `.java` y los archivos de mapeo con extensión `.hbm.xml`. Además, en

el directorio “src”, se encuentran los archivos de configuración de *Hibernate* que indican las características de la base de datos a la que hace referencia el mapeo.

- El directorio “deploy” contiene el archivo .jar general del mapeo llamado “hibernate-gedgapa-3.1.jar”.

Para incluir el mapeo al proyecto y poder trabajar con él, se agregó el archivo .jar “hibernate-gedgapa-3.1.jar” al ambiente de *Maven* por medio del siguiente comando:

```
mvn install:install-file -Dfile=/hibernate-gedgapa-3.1.jar -DartifactId=hibernate-gedgapa -Dversion=1.0.1B -Dpackaging=jar
```

Con esto, el mapeo fue agregado al proyecto y se pudo trabajar con la base de datos mediante sentencias en lenguaje *Java*.

Con las adecuaciones descritas anteriormente, la estructura del sistema quedo conformada de una manera adecuada para el desarrollo de todos los componentes requeridos por el sistema, como interfaces gráficas y clases en *Java*.

5.4 MÉTODO DE PROGRAMACIÓN

La programación o codificación del sistema debe realizarse de manera ordenada, ya que de esta forma se garantiza la calidad del producto, se asegura un avance continuo y se facilita el mantenimiento. Por esta razón es importante tomar en cuenta la forma en que es codificado un sistema.

Existen distintas formas de codificación que dependen del tipo de sistema a desarrollar. Tomando en cuenta que el sistema del PERPAE requería ser una aplicación web, y debido a las características de las tecnologías que se requerían utilizar, se empleó la programación por capas para su codificación.

La programación por capas permite dividir el código según su funcionalidad, lo que facilita la corrección de errores y la reutilización de código. Este tipo de programación es comúnmente utilizada para sistemas web, y tecnologías como *Java*.

A continuación se presenta el modelo de programación que se empleó para la codificación del sistema tomando en cuenta la estructura del proyecto generada con *Maven* y el uso de *Java* e *Hibernate*.

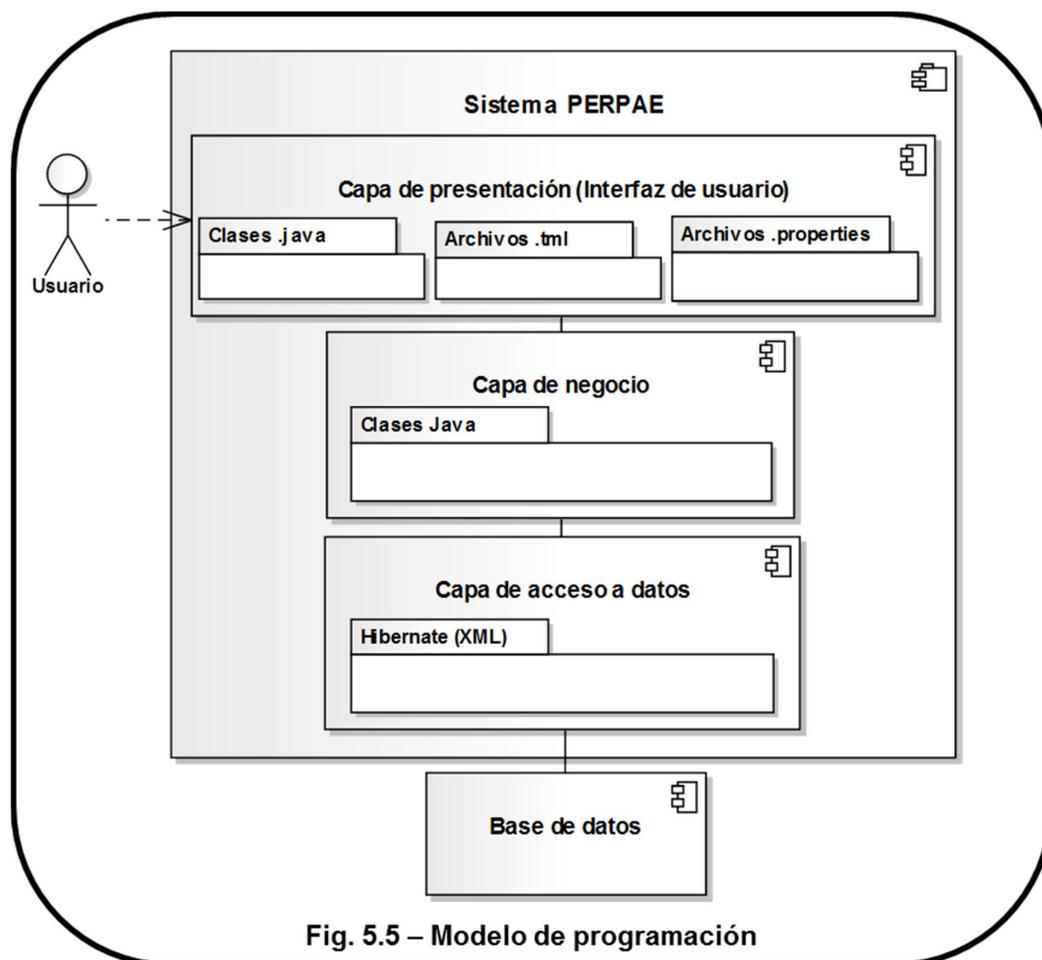


Fig. 5.5 – Modelo de programación

En este diagrama se presentan las siguientes capas:

- **Presentación:** Es la capa por medio de cual, el usuario visualiza e interactúa con el sistema. Contiene la interfaz gráfica y se comunica con la capa de negocio.
- **Negocio:** Es la capa que contiene los programas o procesos que se ejecutan como petición del usuario en la capa de presentación. Contiene las reglas y restricciones del sistema y se comunica con la capa de datos para consultar, almacenar y modificar información.
- **Acceso a datos:** Es la capa que provee acceso a los datos y recibe peticiones de la capa de negocio.

5.5 CODIFICACIÓN

Debido a que el código generado debe ser entendible para garantizar un fácil mantenimiento, se agregaron comentarios indicando la función que realiza cada método implementado. Además, el código de cada archivo fue indentado, es decir, con una sangría adecuada para identificar los bloques de código.

5.5.1 Generación de la Interfaz gráfica

Como se mencionó anteriormente, para la implementación de cada una de las interfaces gráficas se requiere utilizar un archivo .java, un archivo .tml y un archivo .properties. Debido a esto, para cada página o interfaz gráfica, se requirió la generación de estos tres tipos de archivos, tomando en cuenta que el nombre debe ser el mismo para cada página. A continuación se muestra una tabla con el nombre de los archivos generados de acuerdo a cada página del sistema.

Página	Nombre de los archivos .java, . tml y .properties
Actualización de seguros	ActualizarSeguros
Alertas	Alertas
Administración de eméritos	Busqueda
Catálogo de compañías	CatCompania
Catálogo de distinciones	CatDistincion
Catálogo de elementos de difusión	CatElemDifusion
Catálogo de pólizas	CatPoliza
Pólizas contratadas	Companiapoliza
Contactos	Contacto
Cuenta bancaria	CuentaBancaria
Datos personales	DatosPersonales
Designaciones y situación laboral	Designacion
Difusión	Difusion
Distinciones	Distincion
Domicilios	Domicilio
Estudios	Estudio
Gráficas	Grafica
Inicio	Index
Movimientos de padecimientos	MovPadecimiento
Padecimientos	Padecimiento
Reportes	Reporte
Seguro médico	SeguroMedico
Seguros	Seguros
Seguro de vida	SeguroVida
Situación personal	SituacionPersonal

Tabla 5.1 – Páginas del sistema.

Archivos .java de las páginas

Las clases *Java* asociadas a las páginas, contienen los elementos o propiedades que son asociados a los componentes de la página, por tal motivo, son declarados en esta parte. Para declarar estos elementos o propiedades, se utilizó la siguiente sintaxis:

```
@Property  
@Persist  
private TipoElemento nombreElemento;
```

En donde `@Property` indica que el elemento es una propiedad de la página y `@Persist` establece una persistencia del elemento a nivel de página.

Por otra parte, se implementaron los siguientes métodos de las páginas requeridos para procesar información en base a peticiones por parte del usuario:

`public Object initialize ()`: Método utilizado para inicializar las propiedades de la página, las cuales, son asociadas a los elementos gráficos por medio del uso de distintas clases *Java*. En este método también se definen las sentencias para llenado de encabezados e información que requiere ser visualizada en todo momento.

`void onActivate()`: Método que se ejecuta cuando el navegador de internet realiza una petición sobre la página. En este método se definen modelos que sirven para el llenado de listas despegables.

`void setupRender()`: Método que se ejecuta cuando se comienza a cargar la página para ser visualizada.

`Object onSuccess()`: Método que se ejecuta cuando un evento ha sido producido con éxito, como presionar un botón. Contiene las instrucciones para procesar información como respuesta de las peticiones del usuario.

Además de los métodos propios de las páginas, se incluyeron algunos métodos para el manejo de información, tales como:

`void limpiaPagina()`: Método para inicializar elementos de las páginas.

`boolean validaDatos()`: Método para validación de datos antes de ser ingresados en la base de datos.

`void obtieneDatos()`: Método empleado para obtener datos que son mostrados en las páginas.

Archivos .tml de las páginas

Debido a que el código de este tipo de archivos está basado en HTML, su sintaxis está compuesta de etiquetas para representar los componentes de la página.

Algunos de los elementos que fueron empleados para la codificación de las páginas son:

Elemento	Etiqueta	Uso
Sección	div	Utilizados para asignar un conjunto de propiedades a una sección de una página.
Formulario	form	Utilizados para agrupar elementos en una página.
Entrada	Input	Elementos empleados para interactuar con el usuario, principalmente para el registro de información.
Etiqueta	label	Utilizadas para desplegar información de solo lectura.
Bloque	span	Utilizados para agrupar y dar formato a un grupo de elementos.
Hipervínculo	t:actionlink	Empleados para proveer acceso inmediato a un elemento o página.
Mensaje de error	t:errors	Elemento empleado para desplegar mensajes de error.
Condición	t:if	Utilizado para mostrar u ocultar elementos de una página en base a una condición.
Lista desplegable	t:select	Muestra una serie de elementos ordenados para ser seleccionados por el usuario.
Botón	t:submit	Utilizados para realiza peticiones a las páginas.
Área de texto	t:textarea	Permite el manejo de textos amplios.
Campo de texto	t:textfield	Permite el manejo de textos cortos o cadenas.
Tabla	table	Permite agrupar y organizar elementos y datos.

Tabla 5.2 – Elementos gráficos.

También se requirió el uso de propiedades para definir características de algunos de los elementos de las páginas. Algunas de estas propiedades son:

Propiedad	Uso
class	Permite asignar una serie de propiedades a un elemento.
for	Permite establecer una relación entre elementos.
model	Permite asignar un modelo de datos a una lista desplegable.
size	Indica el tamaño de un elemento.
t:id	Permite identificar un elemento con un nombre único.
t:source	Permite asignar una fuente de datos a una tabla de tipo <i>grid</i> .
t:test	Indica el elemento a validar para proceder con una condición.
t:type	Indica un tipo de elemento. Por ejemplo, una tabla de tipo <i>"grid"</i> o una entrada tipo <i>"checkbox"</i> (casilla de verificación).
validate	Permite asignar criterios de validación a un elemento, como la longitud máxima o una expresión regular.
value	Permite asignar el dato con que está asociado un elemento gráfico.

Tabla 5.3 – Propiedades de elementos gráficos.

Algunas de las etiquetas y propiedades utilizadas para representar los elementos comienzan con "t:", esto es debido a que son propias de *Tapestry*.

Para cada elemento de las páginas se asignaron criterios de validación en base al tipo de dato con el que están relacionados. Esto permitió mantener congruencia

entre los tipos de datos de la base de datos y los datos en página. Por otra parte, se asignaron criterios de validación para que la información fuera consistente, es decir, válida y real.

Una característica importante en el desarrollo de las páginas, fue el uso de CSS (*Cascading Style Sheets*), tecnología que permite definir un conjunto de características y asignarlas a varios elementos de forma sencilla. Esto facilita el mantenimiento del sistema y la reutilización de código.

Para utilizar CSS, se definieron en un archivo las clases con las propiedades para cada tipo de elemento, como formularios, títulos y botones; posteriormente se asignó a cada elemento gráfico, la clase requerida por medio de la propiedad "class".

Archivos .properties

En este tipo de archivos, se agregó el código para definir las propiedades de los elementos de las páginas, tales como etiquetas y mensajes. A continuación se presenta la sintaxis empleada para definir los distintos tipos de propiedades:

Para etiquetas asociadas a elementos:
nombreElemento-label=NombreEtiqueta

Para mensajes de validación de elementos:
nombreElemento-criterioValidacion-message=mensajeValidacion

Para mensajes de confirmación:
nombreElemento =MensajeConfirmacion

5.5.2 Codificación de componentes

La interfaz del sistema, además de contar con las páginas antes mencionadas, requirió de una serie de complementos que dieran mayor funcionalidad e hicieran más amigable al sistema. Para esto se utilizó un componente llamado "borde", el cual permitió implementar un marco para las páginas del sistema y generar el menú en el sistema.

Para la implementación del menú del sistema, se generó un archivo .tml en donde se ingresó el código con las opciones del menú por medio de etiquetas. Posteriormente, se implementó una clase *Java* llamada "BordeDgapa" que permite relacionar los elementos del menú con las páginas del sistema.

Para administrar los diferentes tipos de usuarios en el sistema, se implementó la clase "AccessController", la cual permite el manejo de variables durante una sesión en el sistema. Esto permite saber en cualquier momento el usuario y tipo de usuario que ingresa al sistema, lo cual sirve para presentar en página la información de los usuarios que han ingresado al sistema y establecer las restricciones de seguridad en las páginas.

5.5.3 Generación del código de negocio

La codificación de la capa de negocio del sistema es la parte más importante del desarrollo debido a que se implementan los métodos y estructuras que permiten la administración de información.

Los métodos y estructuras que permiten la administración de información fueron manejados como servicios y ubicados en la ruta `\src\main\java\mx\unam\dgapa\perpae\services`. A continuación se describe su implementación y codificación.

Implementación de estructuras

Debido a que los datos deben comunicarse con la interfaz de usuario y la lógica de negocio, se requiere el uso de estructuras que permitan agrupar y manejar de forma sencilla estos datos. Por esta razón se implementaron dos tipos de estructuras en lenguaje *Java* que se describen a continuación.

POJO's (Plain Old Java Object): Son clases en *Java* utilizadas principalmente para establecer estructuras que permiten manejar de forma sencilla los datos mediante *Hibernate*. Estas clases también son empleadas para asignar datos a elementos gráficos de las páginas.

En los POJO's se implementa la serialización debido a que permite convertir un objeto de una clase a un conjunto de bytes para después recuperarlo. Esto facilita el envío de dicho objeto a través de la red.

Los principales métodos que implementan los POJO's son "get" y "set" para cada atributo. Los métodos "get" permiten acceder a los atributos de la clase y los métodos "set" permiten asignar valores a los atributos de la clase.

Las principales POJO's que fueron implementados en el sistema son:

POJO	Uso
BaseModel.java	Manejo de modelos para listas despegables.
CatCompaniasPerpae.java	Manejo de elementos del catálogo de compañías.
CatPolizasPerpae.java	Manejo de elementos del catálogo de pólizas.
CompaniaPolizPerpae.java	Manejo de pólizas contratadas por compañías.
ContactosPerpae.java	Manejo de contactos de eméritos.
CuentaBancariaBitacora.java	Manejo de la bitácora de la cuenta bancaria.
DatosPersonalesBitacora.java	Manejo de la bitácora de datos personales.
DesignacionPerpae.java	Manejo de designaciones de eméritos.
DifusionPerpae.java	Manejo de elementos de difusión de eméritos.
DistincionPerpae.java	Manejo de distinciones de eméritos.
DomicilioPerpae.java	Manejo de domicilios de eméritos.
EmeritoPoliza.java	Manejo de pólizas de eméritos.

EmeritosBitacoraPerpae.java	Manejo de eméritos con información no actualizada.
EstadoSalud.java	Manejo de los estados de salud de eméritos.
NivAcadPerpae.java	Manejo de los estudios y niveles académicos de eméritos.
PersonaNomina.java	Manejo de las personas de nómina.
PersonaPerpae.java	Manejo de eméritos del PERPAE.
SitLaboralBitacora.java	Manejo de la bitácora de situación personal de eméritos.
Visit.java	Manejo de datos de una sesión en el sistema.

Tabla 5.4 – POJO´s utilizados en la codificación.

DTO´s (Data Transfer Object): Son clases para implementar objetos de transferencia de datos, es decir, estructuras que permiten manejar los datos de una forma sencilla y trabajar con distintas estructuras de una base de datos al mismo tiempo.

Para el sistema del PERPAE, se utilizaron DTO´s para manejar en conjunto las estructuras de los diferentes rubros de información, como domicilios o designaciones.

Los DTO´s implementados en el sistema son:

DTO	Uso
BitacotaDTO.java	Manejo de las estructuras asociadas a la bitácora del sistema.
CatPolizasDTO.java	Manejo de las estructuras asociadas a elementos del catálogo de pólizas.
CompaniaPolizaDTO.java	Manejo de las estructuras asociadas a pólizas contratadas por compañías.
CuentaBancariaDTO.java	Manejo de las estructuras asociadas a cuentas bancarias de eméritos.
DatosPersonalesDTO.java	Manejo de las estructuras asociadas a datos personales de personas del PERPAE.
DesignacionDTO.java	Manejo de las estructuras asociadas a las designaciones de eméritos.
DifusionDTO.java	Manejo de las estructuras asociadas a elementos de difusión de eméritos.
DistincionDTO.java	Manejo de las estructuras asociadas a distinciones de eméritos.
DomicilioDTO.java	Manejo de las estructuras asociadas a domicilios de personas del PERPAE.
EstadoSaludDTO.java	Manejo de las estructuras asociadas a estados de salud de personas del PERPAE.
NivelAcademicoDTO.java	Manejo de las estructuras asociadas a niveles académicos de personas del PERPAE.
SeguroDTO.java	Manejo de las estructuras asociadas a pólizas de seguros de eméritos.
SitLaboralDTO.java	Manejo de las estructuras asociadas a la situación laboral de eméritos.

Tabla 5.5 – DTO´s utilizados en la codificación.

Implementación de métodos

Para la implementación de métodos, se emplearon componentes DAO (*Data Access Object* u Objeto de Acceso a Datos), los cuales proporcionan una interfaz entre las páginas del sistema y la base de datos. Los elementos DAO se implementaron por medio de interfaces, por lo que se requirió generar dos archivos .java, uno con los métodos implementados para manejo de la información, y otro de tipo interfaz con los métodos abstractos. Las interfaces implementadas tienen la ventaja de mantener visibles sólo los métodos necesarios para otras clases.

Las interfaces implementadas son:

Interfaz Java	Clase Java	Uso
IBusquedaDAO.java	BusquedaDAO.java	Implementa los métodos para la búsqueda de información en la base de datos mediante <i>Hibernate</i> .
IConstantesDAO.java	ConstantesDAO.java	Implementa los métodos para manejar las constantes del sistema.
IInsercionDAO.java	InsercionDAO.java	Implementa los métodos de inserción de información en la base de datos mediante <i>Hibernate</i> .
IModelosDAO.java	ModelosDAO.java	Permite generar modelos para llenado de listas desplegables.
IHibernate.java	Hibernate.java	Facilita el uso de <i>Hibernate</i>
ICalculosDAO.java	CalculosDAO.java	Implementa métodos para el cálculo de datos.

Tabla 5.6 – Clases DAO utilizadas en la codificación.

Es importante mencionar que la información manejada en el sistema no es eliminada como tal en la base de datos, sino que únicamente se inactiva, por lo que no existe una clase de eliminación de información.

Por otra parte puede observarse que no se implementa una clase de actualización de información, esto es debido a que en *Hibernate*, las sentencias de ingreso y actualización de información son iguales y utilizan el método “persist”, por medio del cual, *Hibernate* identifica si la información es de nuevo ingreso o va a ser actualizada.

5.5.4 Generación de código para acceso a datos

Para el acceso a datos mediante *Hibernate*, se emplearon archivos .xml en donde se implementaron lo *queries* necesarios para el manejo de información. Los archivos utilizados son:

Archivo .xml	Uso
busqueda.xml	Contiene <i>queries</i> para consulta de información.
inserción.xml	Contiene <i>queries</i> de apoyo para la inserción de información.
combos.xml	Contiene <i>queries</i> para llenado de listas desplegables o combos.

Tabla 5.7 – Archivos XML utilizados en la codificación.

Como todo archivo .xml, los archivos para implementar los *queries* emplean etiquetas mediante las cuales se define cada sentencia. De esta manera, los archivos .xml generados cuentan con la siguiente estructura:

```
<queries-hibernate>

    <query_1>
        Sentencias
    </query_1>

    <query_2>
        Sentencias
    </query_2>

</queries-hibernate>
```

Es importante mencionar que se implementaron interfaces en *Java* que facilitaron el manejo de los archivos .xml mencionados. Las interfaces generadas son:

Interfaz <i>Java</i>	Clase <i>Java</i>	Uso
IQuerysBusqueda.java	QuerysBusqueda.java	Acceso a busqueda.xml
IQuerysInsercion.java	QuerysInsercion.java	Acceso a insercion.xml
IQuerysCombos.java	QuerysCombos.java	Acceso a combos.xml

Tabla 5.8 – Interfaces para manejo de *queries*.

Estas clases implementan principalmente un método que hace referencia al nombre y ubicación del archivo asociado. Además permiten a las clases DAO interactuar con los *queries* de los archivos .xml.

Como ejemplo, se presenta la siguiente sentencia que define como se implementa una consulta en la clase BusquedaDAO haciendo referencia a un *query* del archivo busqueda.xml a través de la interfaz IQuerysBusqueda.java:

```
Tpersona tpersona = (Tpersona)session.createSQLQuery(
    querysBusqueda.getString("busca-tpersona-numexp"))
    .addEntity("tpersona", Tpersona.class)
    .setLong("numexp", 120)
    .setMaxResults(1)
    .uniqueResult();
```

En esta consulta se requiere obtener un registro de la tabla Tpersona en base a un RFC. Para esto se declara una entidad del tipo Tpersona, la cual hace referencia a un *query* llamado "busca-tpersona-numexp" asociado a una instancia de la clase IQuerysBusqueda.java llamado "querysBusqueda". En la sentencia presentada se indica que la entidad a llenar es "tpersona" del tipo Tpersona por medio de la clausula. addEntity("tpersona", Tpersona.class). También se indica que se le envía al *query* un parámetro de tipo "Long" llamado "numexp" con el valor 120 por medio de

la cláusula `.setLong("numexp", 120)`. Por último, se indica que el resultado del *query* debe ser único con las cláusulas `.setMaxResults(1)` y `.uniqueResult()`;

El archivo `busqueda.xml` contiene el siguiente *query* llamado “busca-tpersona-numexp”, en el que se puede observar el manejo del parámetro `numexp`:

<busca-tpersona-numexp>

```
select
    tpersona.*
from
    nucleo.tpersona tpersona
where
    tpersona.numexp = :numexp
```

</busca-tpersona-numexp>

5.6 DESARROLLO DE REPORTES Y GRÁFICAS

La implementación de reportes y gráficas requirió de su generación e integración al sistema. A continuación se describen dichas actividades.

5.6.1 Generación de reportes y gráficas

Para la generación de reportes y gráficas se utilizó la herramienta *JasperReports*, la cual permite implementar de manera sencilla los elementos gráficos y las estructuras de datos requeridas.

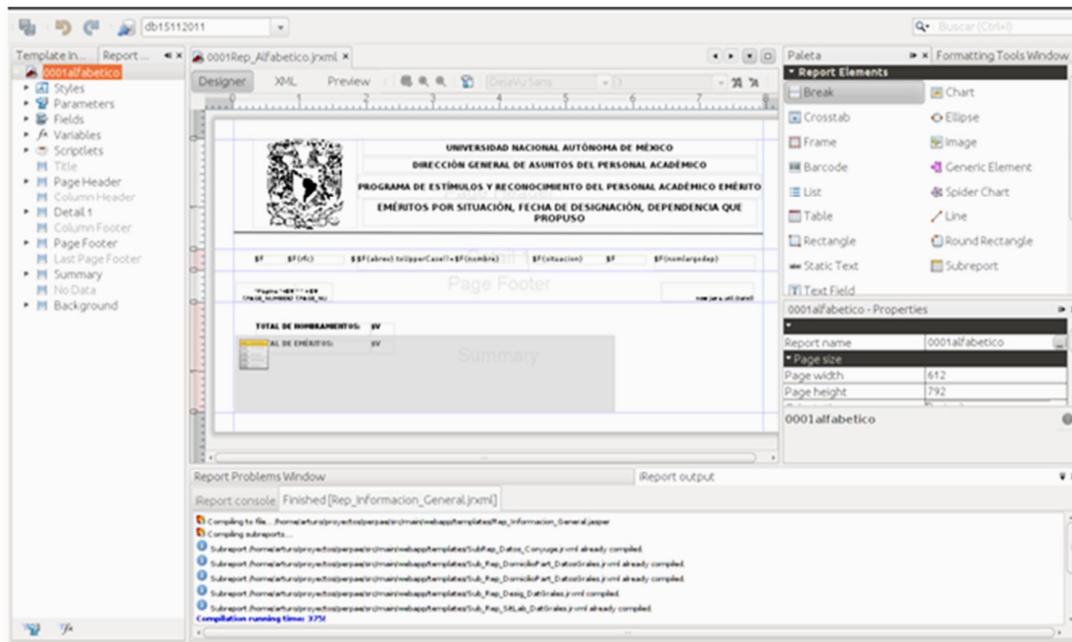


Fig. 5.6 – Interfaz de JasperReports

Generación de reportes

Los reportes generados con *JasperReports*, contienen las siguientes secciones o bandas:

- Título del reporte, presentado únicamente en la primera página del mismo.
- Encabezado del reporte, presentado en todas las páginas.
- Encabezado del detalle, mostrado en cada página del reporte antes del detalle.
- Detalle, conformado por los datos a mostrar en el reporte.
- Pie del detalle, mostrado en cada página del reporte después del detalle.

- Pie de página, mostrado al final de cada página del reporte.
- Resumen, mostrado en la última página del reporte antes del pie de página.

Las secciones de cada reporte pueden contener diferentes elementos gráficos como etiquetas, cuadros de texto e imágenes, los cuales son agregados mediante una barra de herramientas de *JasperReports*.

La generación de los elementos gráficos de los reportes se basó en el diseño de los mismos presentados en el capítulo anterior, tomando en cuenta que el encabezado es el mismo para todos los reportes, a excepción del título del reporte.

Otra parte importante para generar un reporte, es la obtención de datos para llenado del mismo. Para esto se emplearon *queries*, con los que *JasperReports* puede detectar los campos obtenidos de la consulta e incluirlos en el detalle del reporte.

Para la generación de *queries*, se requirió configurar los parámetros de conexión a la base de datos, los cuales incluyen el nombre de la base de datos, el servidor donde se encuentra la base de datos y el usuario y contraseña para ingresar a la misma.

Los *queries* se implementaron en base a las necesidades de cada reporte, considerando los datos que deben mostrarse y aquellos que sirven para cálculo de otros.

Con los *queries* implementados, resultó sencillo presentar los datos en el reporte, ya que por cada dato extraído de la base de datos, *JasperReports* genera un campo que puede ser agregado en las secciones del reporte.

Debido a la estructura de algunos reportes, se requirió realizar agrupaciones de datos en base a distintos criterios, para lo cual se agregaron al reporte secciones llamadas “grupos”.

Una característica importante de *JasperReports*, es que permite dar formato a los diferentes tipos de datos, como fechas y cadenas. Además permite emplear, mediante lenguaje *Java*, funciones y sentencias para manejo de campos y variables. Para los reportes generados, estas funcionalidades se emplearon para dar formato a fechas, concatenar cadenas, implementar sentencias de condición y realizar funciones sobre cadenas, como la conversión a mayúsculas o la obtención de subcadenas.

Debido a que los reportes requieren un resumen con información específica y precisa, se hizo uso de subreportes, los cuales son reportes incluidos en otros reportes.

Generación de gráficas

Las gráficas generadas fueron implementadas de la misma manera que reportes, con la diferencia de que presentaron un diseño distinto e incluyeron elementos de tipo “gráfica” para representar los datos de una manera más visual.

5.6.2 Integración de reportes y gráficas al sistema

Para la integración de los reportes y gráficas al sistema se realizó lo siguiente:

- Se creó un directorio llamado “templates” en la ubicación `\src\main\webapp\templates` del proyecto en donde se ingresaron las gráficas y reportes generados.
- Se creó una pantalla llamada “Print” para mostrar los reportes y gráficas al usuario.
- Se creó una clase llamada `PrintJasper.java` para generar reportes y gráficas por medio de *JasperReports* y presentarlos en pantalla de acuerdo a un formato seleccionado.
- Se implementaron en las páginas “reportes” y “gráficas”, los métodos necesarios para generar los reportes y gráficas requeridos por el usuario.

CAPÍTULO VI

PRUEBAS E IMPLANTACIÓN

En este capítulo se describen las pruebas realizadas sobre el sistema de información desarrollado, las cuales ayudaron a identificar las mejoras a realizar y garantizaron que el sistema cumpliera con las características requeridas.

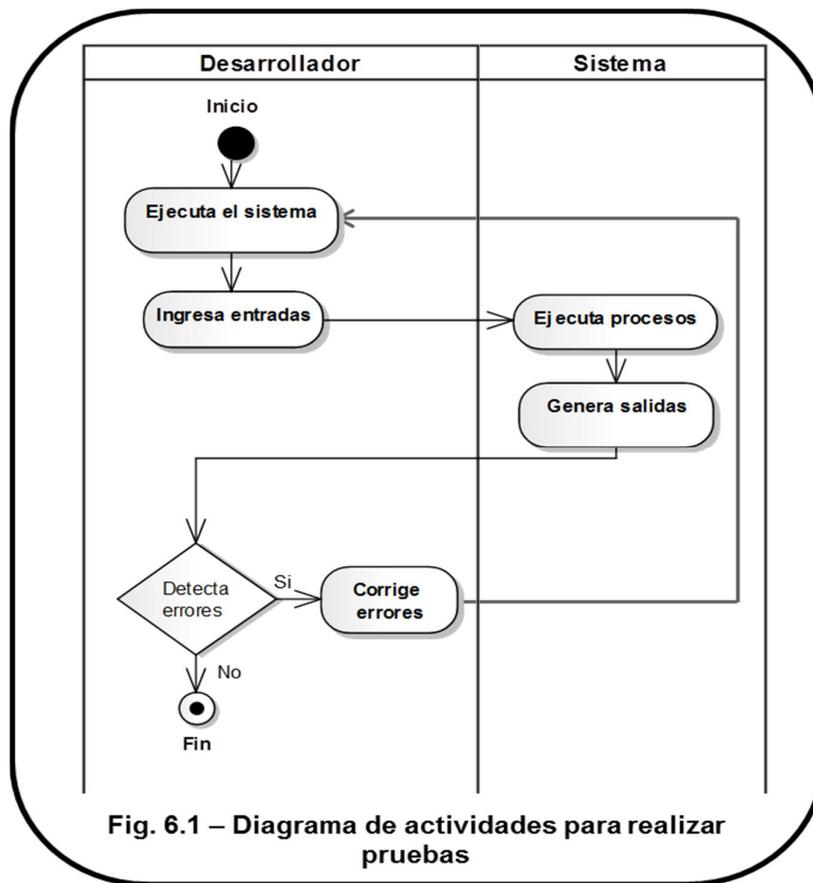
En este capítulo también se describe el proceso de implantación, mediante el cual, se puso en operación el sistema de información del PERPAE en el hardware adecuado para su uso.

6.1 PRUEBAS

La realización de pruebas es una tarea muy importante durante el ciclo de vida de un software debido a que permite identificar fallas y verificar su funcionamiento.

Para el sistema de información del PERPAE se realizaron diferentes tipos de pruebas para conocer su calidad, identificar y corregir errores, y evaluar sus componentes.

Las pruebas realizadas permitieron evaluar salidas en base a entradas y detectar errores en el sistema de información. El proceso para realizar pruebas se puede representar con el siguiente diagrama de actividades:



A continuación se describen las distintas pruebas realizadas sobre el sistema de información del PERPAE.

6.1.1 Pruebas funcionales

Las pruebas funcionales son implementadas sobre un sistema de información para verificar que este realiza las operaciones y procesos que se definieron durante su diseño.

Para realizar las pruebas funcionales, se utilizaron los casos de uso previamente generados, en los cuales se definieron escenarios derivados de los requerimientos funcionales del sistema de información.

Algunos de los procesos validados mediante las pruebas funcionales son: alta de información, baja de información, consulta de información, actualización de información, generación de reportes, generación de gráficas y generación de alertas.

6.1.2 Pruebas de caja blanca

Las pruebas de caja blanca son implementadas sobre un software para determinar si cada componente o parte del código, se ejecuta de manera adecuada.

Estas pruebas se realizaron sobre el sistema de información para asegurar que todos los caminos posibles del código se ejecutaran de manera correcta. Para estas pruebas se consideró lo siguiente:

- Las rutinas de error que pueden ejecutarse.
- Las validaciones de datos que debe realizar el sistema para asegurar que la información cuente con un formato adecuado.
- Los mensajes que debe generar el sistema ante diferentes escenarios.

Estas pruebas se realizaron por cada página del sistema, ya que de esta manera se seccionó el código de una manera adecuada para identificar errores específicos de cada proceso relacionado con cada conjunto de información.

Para efectuar estas pruebas, se ingresaron datos de entrada que incluían información válida y no válida, así como datos con formatos correctos e incorrectos. Esto permitió identificar las secciones de código que requerían ajustes y las validaciones que se necesitaban implementar.

Por otra parte se implementaron, en una base de datos de prueba, triggers para observar el funcionamiento del sistema al realizar acciones de inserción y actualización de información. Esto permitió identificar errores al insertar y actualizar datos.

6.1.3 Pruebas de integración

Estas pruebas se realizaron para detectar errores producidos por la combinación de componentes de software del sistema.

Estas pruebas resultaron de gran apoyo para verificar el funcionamiento del sistema al integrar tecnologías como *Hibernate* o *Subversion*. También permitieron detectar errores al integrar diferentes páginas y módulos del sistema.

6.1.4 Pruebas de seguridad

Debido a que el sistema de información debe proveer seguridad, se implementaron este tipo de pruebas mediante las cuales se verificó y validó que únicamente los usuarios autorizados pudieran acceder al sistema y realizaran las funciones permitidas.

Para estas pruebas se utilizaron datos válidos y no válidos de usuarios para ingresar al sistema, lo que permitió observar el nivel de seguridad del mismo y corregir las fallas detectadas.

6.1.5 Pruebas de usabilidad

Estas pruebas fueron realizadas para determinar si el sistema era fácil de utilizar desde el punto de vista de los usuarios. También permitieron identificar la complejidad, amigabilidad, visibilidad, apariencia y lógica de los elementos y funciones de cada pantalla del sistema.

Para estas pruebas, los usuarios realizaron procesos específicos con información real, lo que facilitó la comprensión del sistema y ayudó a determinar las mejoras que debían realizarse tomando en cuenta la perspectiva de los usuarios.

6.1.6 Pruebas de rendimiento

Las pruebas de rendimiento permiten conocer la rapidez con que un sistema realiza diversas tareas bajo ciertas condiciones.

Para la realización de estas pruebas se ejecutaron procesos de alta, baja, actualización y consulta de información con diferentes volúmenes de datos, con lo que se determinó el tiempo que el sistema empleaba para realizar dichos procesos.

Debido a que el sistema respondió de manera adecuada bajo diversas condiciones y mostró un buen rendimiento, no se requirió realizar ajustes sobre sus componentes.

6.1.7 Pruebas de compatibilidad

Debido a que el sistema desarrollado es una herramienta web que requiere el uso de un navegador de internet, se realizaron pruebas de compatibilidad para determinar si el sistema era compatible con los navegadores de internet más utilizados.

Algunos de los navegadores utilizados para realizar estas pruebas son: *Internet Explorer*, *Mozilla Firefox* y *Google Chrome*.

6.2 IMPLANTACIÓN

El proceso de implantación de un software engloba una serie de actividades cuyo fin es instalar el sistema en el hardware indicado para su uso.

Debido a que el sistema de información del PERPAE es un sistema web, no se requirió de su instalación en los equipos de cómputo donde es utilizado, sino que fue alojado en un servidor que proporciona acceso al sistema.

Para la implantación del sistema se generó un archivo WAR (*Web Application Archive* o archivo de aplicación web), el cual permite ejecutar aplicaciones en *Java* distribuyendo los componentes que conforman dicha aplicación, como clases *Java*, archivos XML y archivos HTML.

Para la generación del archivo WAR, se empleó el siguiente comando de *Maven*:

```
mvn package
```

El archivo WAR generado, así como las estructuras de datos del sistema PERPAE, fueron alojados en servidores con las siguientes características:

- 2 procesadores Intel(R) Xenon(TM)
- CPU E5620 con 2.40 GHz
- 16 GB RAM
- 260 GB de disco duro con conexión de tipo IDE (*Integrated Drive Electronics* o dispositivo electrónico integrado)
- 6 tarjetas de red
- Sistema operativo Ubuntu 10.04 Lucid

Para que el sistema fuera vinculado al sistema GeDGAPA y pudiera ser utilizado, se ligó el archivo WAR generado a una opción llamada "PERPAE" en la interfaz gráfica del sistema GeDGAPA.

CONCLUSIONES

El desarrollo del sistema de información presentado en este trabajo, requirió del uso de aptitudes y conocimientos que permitieron obtener una herramienta útil y que cubre las necesidades de los usuarios para los que fue diseñado.

En cada una de las fases del desarrollo del sistema, se realizaron actividades importantes que otorgaron al sistema calidad y funcionalidad.

En la fase de preparación, se definieron los objetivos del sistema y se estableció la planeación a seguir. Las actividades realizadas en esta etapa permitieron consolidar una base para el desarrollo del sistema y obtener una amplia perspectiva del mismo.

Durante las actividades de análisis y diseño se definieron a detalle las características con que el sistema debía contar, además de definir un modelado del mismo.

Durante las actividades de desarrollo, se realizó la construcción del sistema en base al análisis y diseño del mismo, lo que trajo como resultado la obtención una herramienta en la que se proyectaron los requerimientos previamente definidos.

Las pruebas realizadas permitieron hacer del sistema una herramienta de calidad, además de permitir identificar errores para posteriormente corregirlos.

Al finalizar con el desarrollo del sistema de información del PERPAE, se puede mencionar que se cumplieron satisfactoriamente los objetivos planteados al inicio de este trabajo, ya que se logró implementar un sistema de apoyo en la administración de información del PERPAE, el cual cuenta con las características suficientes para ser gestionado tecnológicamente por la DGAPA y permitir su fácil mantenimiento.

Además de cubrir los objetivos planteados inicialmente, el desarrollo del sistema del PERPAE trajo consigo las siguientes ventajas al personal de la UNAM y en especial a miembros de la DGAPA:

- El sistema cuenta con un mecanismo de alertas, convirtiéndolo en el primer sistema desarrollado en la DGAPA con esta característica, lo cual servirá de base para el desarrollo de otros sistemas.
- El sistema desarrollado cuenta con características que le permiten ser una herramienta portable y escalable, lo que facilita y propicia la extensión de su uso.
- El sistema permitió optimizar y automatizar procesos que anteriormente se realizaban de forma manual, lo cual disminuyó tiempos de trabajo y facilitó la realización de actividades.
- El sistema desarrollado es amigable y fácil de usar, lo que propició el sencillo aprendizaje por parte de los usuarios.

El sistema desarrollado fue evaluado de manera benéfica por parte de los usuarios y del personal directivo, además de ser valorado como un sistema integral, útil, amigable y con características innovadoras.

A pesar de que la metodología seleccionada para el desarrollo del sistema no se siguió en su totalidad debido a diversos factores, como la forma de trabajo de la DGAPA, el sistema resultó ser una herramienta de calidad, cuyo éxito se basó principalmente en el trabajo, la comunicación, la responsabilidad y la organización.

ANEXOS

DICcionario DE DATOS

Estructuras del esquema “catálogos” utilizadas para dar soporte al sistema de información del PERPAE

cardies : Catálogo de áreas, disciplinas y especialidades.

Nombre	Tipo de dato	Nulo	Descripción
cveardies(PK)	Integer	No	Clave del catálogo cardies
cveardiesparent(FK)	Integer	No	Clave del catálogo cardies
cvenomardies(FK)	Integer	No	Clave del catálogo cnomardies
cvetipoardies(FK)	Integer	No	Clave del catálogo ctipoardies
Estado	Char(1)	No	Estado del catálogo cardies

casuntos : Catálogo de asuntos (PAPIIT, PAPIME, INFOCAB, CURSOS, PERPAE, etc.)

Nombre	Tipo de dato	Nulo	Descripción
cveasunto(PK)	Integer	No	Clave del catálogo casuntos
Descasunto	Varchar(255)	No	Descripción
Estado	Char(1)	No	Estado del catálogo casuntos

ccategorias : Catálogo de categorías de acuerdo al catálogo presupuestal.

Nombre	Tipo de dato	Nulo	Descripción
cvecateg(PK)	Integer	No	Clave del catálogo ccategorias
Codcateg	Varchar(10)	No	Código de acuerdo al catálogo presupuestal
Nomcateg	Varchar(255)	No	Nombre de la categoría
Estado	Char(1)	No	Estado del catálogo ccategorias

cdependencias : Catálogo de dependencias.

Nombre	Tipo de dato	Nulo	Descripción
cvedep(PK)	Integer	No	Clave del catálogo cdependencias
Coddep	Varchar(10)	No	Código de la dependencia
Nomcortodep	Varchar(255)	No	Nombre corto de la dependencia
Nomlargodep	Varchar(255)	No	Nombre largo de la dependencia
Sigladep	Varchar(50)	Si	Sigla de la dependencia
Estado	Char(1)	No	Estado del catálogo cdependencias
Articulo	Varchar(10)	Si	Artículo
Preposición	Varchar(10)	Si	Preposición

cdepsubstiasun : Catálogo que relaciona dependencias con los subsistemas y asuntos.

Nombre	Tipo de dato	Nulo	Descripción
cvedepsustiasun(PK)	Integer	No	Clave del catálogo cdepsustiasun
cvesustiasun(FK)	Integer	No	Clave del catálogo csustiasun
cvedep(FK)	Integer	No	Clave del catálogo cdependencias
Estado	Char(1)	No	Estado del catálogo cdepsustiasun

cdispnivacad : Catálogo de disponibilidad de niveles académicos por asunto.

Nombre	Tipo de dato	Nulo	Descripción
cveasunto(FK)	Integer	No	Clave del catálogo casuntos
cvenivacad	Integer	No	Clave del catálogo cnivacad

cedocivil : Catálogo de estados civiles.

Nombre	Tipo de dato	Nulo	Descripción
cveedocivil(PK)	Integer	No	Clave del catálogo cedocivil
cveedocivilparent(FK)	Integer	No	Clave del catálogo cedocivil
descedocivil	Varchar(255)	No	Descripción del estado civil
estado	Char(1)	No	Estado del catálogo cedocivil

centregion : Catálogo de entidades y regiones.

Nombre	Tipo de dato	Nulo	Descripción
cveentregion(PK)	Integer	No	Clave del catálogo centregion
cveentregionparent(FK)	Integer	No	Clave del catálogo centregion
cvetipoentregion(FK)	Smallint	No	Clave del catálogo ctipoentregion
nomentregion	Varchar(255)	No	Nombre de la región
siglaentregion	Varchar(255)	No	Sigla de la región
estado	Char(1)	No	Estado del catálogo centregion

cetapaconvasun : Catálogo que relaciona las etapas con las versiones de las convocatorias y asuntos.

Nombre	Tipo de dato	Nulo	Descripción
cveetapaconvasun(PK)	Integer	No	Clave del catálogo cetapaconvasun
cveetapaconvasunparent(FK)	Integer	No	Clave del catálogo cetapaconvasun
cveasunto(FK)	Integer	No	Clave del catálogo casuntos
cveconv(FK)	Integer	No	Clave del catálogo cconvocatorias
cveetapa(FK)	Integer	No	Clave del catálogo cetapas
Estado	Char(1)	No	Estado del catálogo cetapaconvasun

cinstituciones : Catálogo de instituciones.

Nombre	Tipo de dato	Nulo	Descripción
cveinst(PK)	Integer	No	Clave del catálogo cinstituciones
cveinstparent(FK)	Integer	No	Clave del catálogo cinstituciones
Nominst	Varchar(255)	No	Nombre de la institución
siglainst	Varchar(50)	Si	Sigla de la institución
Estado	Char(1)	No	Estado del catálogo cinstituciones

cnivacad : Catálogo de niveles académicos.

Nombre	Tipo de dato	Nulo	Descripción
cvenivacad(PK)	Integer	No	Clave del catálogo cnivacad
descnivacad	Varchar(255)	No	Descripción del nivel académico
Estado	Char(1)	No	Estado del catálogo cnivacad

cnomardies : Catálogo de nombres de áreas, disciplinas y especialidades.

Nombre	Tipo de dato	Nulo	Descripción
cvenomardies(PK)	Integer	No	Clave del catálogo cnomardies
nomardies	Varchar(255)	Si	Nombre del área, disciplina o especialidad
Estado	Char(1)	No	Estado del catálogo cnomardies

croles : Catálogo de roles de acuerdo a jerarquía.

Nombre	Tipo de dato	Nulo	Descripción
cverol(PK)	Integer	No	Clave del catálogo croles
cverolparent(FK)	Integer	No	Clave del catálogo croles
Descrol	Varchar(255)	No	Descripción del rol
Estado	Char(1)	No	Estado del catálogo croles

crolesasun : Catálogo que relaciona los roles y los asuntos donde es utilizado.

Nombre	Tipo de dato	Nulo	Descripción
cverolasun(PK)	Integer	No	Clave del catálogo crolesasun
cveasunto(FK)	Integer	No	Clave del catálogo casuntos
cverol(FK)	Integer	No	Clave del catálogo croles
Estado	Char(1)	No	Estado del catálogo crolesasun

csubsistasun : Relación de subsistemas con los asuntos.

Nombre	Tipo de dato	Nulo	Descripción
cvesubsistasun(PK)	Integer	No	Clave del catálogo csubsistasun
cveasunto(FK)	Integer	No	Clave del catálogo casuntos
cvesubsistema(FK)	Integer	No	Clave del catálogo csubsistema
Estado	Char(1)	No	Estado del catálogo csubsistasun

csubsistemas : Catálogo de subsistemas.

Nombre	Tipo de dato	Nulo	Descripción
cvesubsistema(PK)	Integer	No	Clave del catálogo csubsistemas
descsubsistema	Varchar(255)	No	Descripción del subsistema
Estado	Char(1)	No	Estado del catálogo csubsistemas

ctipoardies : Catálogo de tipos de área, disciplina y especialidad.

Nombre	Tipo de dato	Nulo	Descripción
cvetipoardies(PK)	Integer	No	Clave del catálogo ctipoardies
desctipoardies	Varchar(255)	No	Descripción del tipo de área, disciplina o especialidad
Estado	Char(1)	No	Estado del catálogo ctipoardies

ctipodom : Catálogo de tipos de domicilio.

Nombre	Tipo de dato	Nulo	Descripción
cvetipodom(PK)	Integer	No	Clave del catálogo ctipodom
desctipodom	Varchar(255)	No	Descripción del tipo de domicilio
Estado	Char(1)	No	Estado del catálogo ctipodom

ctipotel : Catálogo de tipos de teléfono.

Nombre	Tipo de dato	Nulo	Descripción
cvetipotel(PK)	Integer	No	Clave del catálogo ctipotel
desctipotel	Varchar(255)	No	Descripción del tipo de teléfono
Estado	Char(1)	No	Estado del catálogo ctipotel

Estructuras del esquema “nómina” utilizadas para dar soporte al sistema de información del PERPAE.

tnomina : Tabla de académicos en nómina.

Nombre	Tipo de dato	Nulo	Descripción
Rfc	Char(10)	No	RFC
homohda	Char(3)	No	Homoclave de hacienda
Apepat	Char(40)	No	Apellido paterno
apepmat	Char(40)	Si	Apellido materno
Nombre	Char(50)	No	Nombre
Sueldo	Char(9)	No	Sueldo
categoría	Char(14)	No	Categoría
cvecategoria	Char(5)	No	Clave de la categoría
Nivcateg	Char(2)	No	Nivel de la categoría
numplaza	Char(5)	No	Número de plaza
Cveprog	Char(2)	No	Clave del programa
cvesubprog	Char(2)	No	Clave del subprograma
Cvedep	Char(5)	No	Clave de la dependencia
cvepartida	Char(3)	No	Clave de partida
cvetipoejer	Char(1)	No	Clave tipo de ejercicio
Digctrl	Char(1)	No	Dig. Control
antseradmo	Char(2)	No	Antigüedad servicio administrativo
porcompantadmo	Char(4)	No	
feciniultnomacad	Char(4)	No	Fecha inicio último nombramiento académico
antseracad	Char(2)	No	Antigüedad servicio académico
porcompantacad	Char()	No	
Horas	Char(3)	No	Horas
Genero	Char(1)	No	Género
feciniultnomadmo	Char(8)	No	Fecha inicio último nombramiento administrativo
quincena	Char(6)	No	Quincena
Numexp	Integer	Si	Número de expediente
cvecateg	Integer	No	Clave de la categoría
cvedepsubdep	Integer	No	Clave de la dependencia o subdependencia

Estructuras del esquema “núcleo” utilizadas para dar soporte al sistema de información del PERPAE

tdomicilio : Tabla de domicilios de personas.

Nombre	Tipo de dato	Nulo	Descripción
iddom(PK)	Integer	No	Identificador de la tabla tdomicilio
idper(FK)	Integer	No	Identificador de la tabla tpersona
cvetipodom(FK)	Integer	No	Clave del catálogo ctipodom
cveentregion(FK)	Integer	No	Clave del catálogo centregion
Calle	Varchar(255)	Si	Calle del domicilio
Col	Varchar(255)	Si	Colonia del domicilio
Pob	Varchar(255)	Si	Población del domicilio
Codpos	Char(10)	Si	Código postal del domicilio
Estado	Char(1)	No	Estado de la tabla tdomicilio

tdomrelasun : Tabla que relaciona domicilios con asuntos.

Nombre	Tipo de dato	Nulo	Descripción
iddomrelasun(PK)	Integer	No	Identificador de la tabla tdomrelasun
iddom(FK)	Integer	No	Identificador de la tabla tdomicilio
cveasunto(FK)	Integer	No	Clave del catálogo casuntos
fecregdomrelasun	Smalldatetime	No	Fecha de registro
estado	Char(1)	No	Estado de la tabla tdomrelasun

tedocivil : Tabla de estado civil de personas.

Nombre	Tipo de dato	Nulo	Descripción
idedocivil(PK)	Integer	No	Identificador de la tabla tedocivil
idper(FK)	Integer	No	Identificador de la tabla tpersona
cveedocivil(FK)	Integer	No	Clave del catálogo cedocivil
fecreg	Smalldatetime	No	Fecha de registro

tedocivilasun : Tabla que relaciona estados civiles con asuntos.

Nombre	Tipo de dato	Nulo	Descripción
idedocivilasun(PK)	Integer	No	Identificador de la tabla tedocivilasun
idedocivil(FK)	Integer	No	Identificador de la tabla tedocivil
cveasunto(FK)	Integer	No	Clave del catálogo casuntos
fecregdomrelasun	Smalldatetime	No	Fecha de registro

temail : Tabla de correos electrónicos de personas.

Nombre	Tipo de dato	Nulo	Descripción
idemail(PK)	Integer	No	Identificador de la tabla temail
idper(FK)	Integer	No	Identificador de la tabla tpersona
email	Varchar(255)	No	Email
fecregemail	Smalldatetime	No	Fecha de registro
estado	Char(1)	No	Estado de la tabla ttrivacad

tnacionalidad : Tabla que relaciona nacionalidades con regiones.

Nombre	Tipo de dato	Nulo	Descripción
idnacionalidad(PK)	Integer	No	Identificador de la tabla tnacionalidad
cveentregion(FK)	Integer	No	Clave del catálogo centregion
nacionalidad	Varchar(255)	Si	Nacionalidad

tnivacadrelasun : Relación de los niveles académicos de las personas con los asuntos.

Nombre	Tipo de dato	Nulo	Descripción
idnivacadrelasun(PK)	Integer	No	Identificador de la tabla tnivacadrelasun
idnivacad(FK)	Integer	No	Identificador de la tabla tnivacad
cveasunto(FK)	Integer	No	Clave del catálogo casuntos
fecregnivacad	Smalldatetime	No	Fecha de registro

tpersona : Tabla de personas.

Nombre	Tipo de dato	Nulo	Descripción
idper(PK)	Integer	No	Identificador de la tabla tpersona
curp	Char(18)	Si	CURP
rfc	Char(13)	Si	RFC
numexp	Integer	Si	Número de expediente
aopat	Varchar(80)	No	Apellido paterno
apemat	Varchar(80)	No	Apellido materno
nombre	Varchar(80)	No	Nombre
genero	Char(1)	No	Género
cveentregion(FK)	Integer	No	Clave del catálogo centregion
fecnac	Smalldatetime	Si	Fecha de nacimiento

trolper : Tabla que relaciona roles con personas.

Nombre	Tipo de dato	Nulo	Descripción
idrolper(PK)	Integer	No	Identificador de la tabla trolper
idper(FK)	Integer	No	Identificador de la tabla tpersona
cverolasun(FK)	Integer	No	Clave del catálogo casuntos
fecregtipopart	Smalldatetime	Si	Fecha de registro

ttelefono : Tabla de teléfonos de personas.

Nombre	Tipo de dato	Nulo	Descripción
idtel(PK)	Integer	No	Identificador de la tabla ttelefono
idper(FK)	Integer	No	Identificador de la tabla tpersona
cvetipotel(FK)	Integer	No	Clave del catálogo ctipotel
numero	Varchar(255)	No	Número telefónico
fecregtipopart	Smalldatetime	No	Fecha de registro

ttelrelasun : Tabla que relaciona teléfonos con asuntos.

Nombre	Tipo de dato	Nulo	Descripción
idtelrelasun(PK)	Integer	No	Identificador de la tabla ttelrelasun
idtel(FK)	Integer	No	Identificador de la tabla ttelefono
cveasunto(FK)	Integer	No	Clave del catálogo casuntos
fecregtipopart	Smalldatetime	No	Fecha de registro
estado	Char(1)	No	Estado de la tabla ttelrelasun

Estructuras propias del sistema de información del PERPAE

cambitodist : Catálogo de ámbitos de distinciones (nacional, internacional o extranjero).

Nombre	Tipo de dato	Nulo	Descripción
cveambitodist(PK)	Integer	No	Clave del catálogo cambitodist
descambitodist	Varchar(255)	No	Descripción del ámbito
estado	Char(1)	No	Estado del catálogo cambitodist

ccompanias : Catálogo de compañías bancarias y de seguros.

Nombre	Tipo de dato	Nulo	Descripción
cvecompania(PK)	Integer	No	Clave del catálogo ccompanias
cvetipocompania(FK)	Integer	No	Clave del catálogo ctipocompania
desccompania	Varchar(255)	No	Descripción de la compañía
estado	Char(1)	No	Estado del catálogo ccompanias

cdesignados : Catálogo de tipos de designación (profesor o investigador emérito).

Nombre	Tipo de dato	Nulo	Descripción
cvedesig(PK)	Integer	No	Clave del catálogo cdesignados
descdesig	Varchar(255)	No	Descripción del tipo de designación
estado	Char(1)	No	Estado del catálogo cdesignados

cdistinciones : Catálogo de distinciones.

Nombre	Tipo de dato	Nulo	Descripción
cvedist(PK)	Integer	No	Clave del catálogo cdistinciones
nomdist	Varchar(100)	No	Nombre de la distinción
abrevdist	Varchar(255)	No	Abreviatura de la distinción
estado	Char(1)	No	Estado del catálogo cdistinciones

cedosalud : Catálogo de estados de salud.

Nombre	Tipo de dato	Nulo	Descripción
cveedosalud(PK)	Integer	No	Clave del catálogo cedosalud
descedosalud	Varchar(255)	No	Descripción del estado de salud
estado	Char(1)	No	Estado del catálogo cedosalud

celemaudit : Catálogo de elementos auditables para la generación de alertas.

Nombre	Tipo de dato	Nulo	Descripción
cvecelemaudit(PK)	Integer	No	Clave del catálogo celemaudit
descelemaudit	Varchar(255)	No	Descripción del elemento
tiempoalert	Integer	Si	Tiempo para generar la alerta
Estado	Char(1)	Si	Estado del catálogo celemaudit

cleyisste : Catálogo de tipos de ley del ISSSTE.

Nombre	Tipo de dato	Nulo	Descripción
cveleyisste(PK)	Integer	No	Clave del catálogo cleyisste
descleyisste	Varchar(255)	No	Descripción de la ley del ISSSTE
Estado	Char(1)	No	Estado del catálogo cleyisste

cpolizacomp : Relación de pólizas que son compatibles.

Nombre	Tipo de dato	Nulo	Descripción
cvepolizacomp(PK)	Integer	No	Clave del catálogo cpolizacomp
cvapoliza	Integer	No	Clave del catálogo cpolizas
cvepoliza2	Integer	No	Clave del catálogo cpolizas

cpolizas : Catálogo de pólizas.

Nombre	Tipo de dato	Nulo	Descripción
cvepoliza(PK)	Integer	No	Clave del catálogo cpolizas
cvetipopoliza(FK)	Integer	No	Clave del catálogo ctipopoliza
descpoliza	Varchar(255)	No	Descripción de la póliza
estado	Char(1)	No	Estado del catálogo cpolizas

csitciclosoli : Catálogo de situaciones de los ciclos de la solicitud.

Nombre	Tipo de dato	Nulo	Descripción
cvesitciclosoli(PK)	Integer	No	Clave del catálogo csitciclosoli
cvesitciclosoliparent(FK)	Integer	No	Clave del catálogo csitciclosoli
cvetipoflujoasun(FK)	Integer	No	Clave del catálogo ctipoflujoasun
descsitciclosoli	Varchar(255)	No	Descripción de la situación.
estado	Char(1)	No	Estado del catálogo csitciclosoli

csittipart : Catálogo de situaciones del tipo de participación.

Nombre	Tipo de dato	Nulo	Descripción
cvesittipart(PK)	Integer	No	Clave del catálogo csittipart
cvesittipartparent(FK)	Integer	No	Clave del catálogo csittipart
descsittipart	Varchar(255)	No	Descripción de la situación del tipo de participación
estado	Char(1)	No	Estado del catálogo csittipart

ctipoasocarea : Catálogo de tipos de área asociado a una disciplina o especialidad.

Nombre	Tipo de dato	Nulo	Descripción
cvetipoasocarea(PK)	Integer	No	Clave del catálogo ctipoasocarea
descctipoasocarea	Varchar(255)	No	Descripción de la asociación de área
estado	Char(1)	No	Estado del catálogo ctipoasocarea

ctipocompania : Catálogo de tipos de compañías (bancaria y de seguros).

Nombre	Tipo de dato	Nulo	Descripción
cvetipocompania(PK)	Integer	No	Clave del catálogo ctipocompania
desctipocompania	Varchar(255)	No	Descripción del tipo de compañía
estado	Char(1)	No	Estado del catálogo ctipocompania

ctipoelemdif : Catálogo de tipos de elementos de difusión.

Nombre	Tipo de dato	Nulo	Descripción
cvetipoelemdif(PK)	Integer	No	Clave del catálogo ctipoelemdif
desctipoelemdif	Varchar(255)	No	Descripción del tipo de elemento de difusión
estado	Char(1)	No	Estado del catálogo ctipoelemdif

ctipoestim : Catálogo de tipos de estímulos económicos (semestral, catedra II , mensual).

Nombre	Tipo de dato	Nulo	Descripción
cvetipoestim(PK)	Integer	No	Clave del catálogo ctipoestim
desctipoestim	Varchar(255)	No	Descripción del tipo de estímulo
estado	Char(1)	No	Estado del catálogo ctipoestim

ctipogestor : Catálogo de tipos de gestores en el sistema (administrador, usuario de consulta).

Nombre	Tipo de dato	Nulo	Descripción
cvetipogestor(PK)	Integer	No	Clave del catálogo ctipogestor
desctipogestor	Varchar(255)	No	Descripción del tipo de gestor
estado	Char(1)	No	Estado del catálogo ctipogestor

ctipohrsasig : Catálogo de tipos de horas de asignatura.

Nombre	Tipo de dato	Nulo	Descripción
cvetipohrsasig(PK)	Integer	No	Clave del catálogo ctipohrsasig
desctipohrsasig	Varchar(255)	No	Descripción del tipo de horas de asignatura
estado	Char(1)	No	Estado del catálogo ctipohrsasig

ctipoobs : Catálogo de tipo de observaciones (Datos personales, contactos, etc.).

Nombre	Tipo de dato	Nulo	Descripción
cvetipoobs(PK)	Integer	No	Clave del catálogo ctipoobs
desctipoobs	Varchar(255)	No	Descripción del tipo de observación
estado	Char(1)	No	Estado del catálogo ctipoobs

ctipopagomovpadeci : Catálogo de tipos de pago de los movimientos del padecimiento (cheque, cuenta de débito)

Nombre	Tipo de dato	Nulo	Descripción
cvetipopagomovpadeci(PK)	Integer	No	Clave del catálogo ctipopagomovpadeci
desctipopagomovpadeci	Varchar(255)	No	Descripción del tipo de pago
estado	Char(1)	No	Estado del catálogo ctipopagomovpadeci

ctipopoliza : Catálogo de tipos de pólizas (vida, gastos médicos).

Nombre	Tipo de dato	Nulo	Descripción
cvetipopoliza(PK)	Integer	No	Clave del catálogo ctipopoliza
desctipopoliza	Varchar(255)	No	Descripción del tipo de póliza
estado	Char(1)	No	Estado del catálogo ctipopoliza

ctiporoldif : Catálogo de tipos de rol que desempeña el participante en el elemento de difusión.

Nombre	Tipo de dato	Nulo	Descripción
cvetiporoldif(PK)	Integer	No	Clave del catálogo ctiporoldif
desctiporoldif	Varchar(255)	No	Descripción del tipo de rol
estado	Char(1)	No	Estado del catálogo ctiporoldif

tasocarea : Tabla que relaciona áreas con el concepto de disciplinas o especialidad de estudios.

Nombre	Tipo de dato	Nulo	Descripción
idasocarea(PK)	Integer	No	Identificador de la tabla tasocarea
idnivacadtipopart(FK)	Integer	No	Identificador de la tabla tnvicadtipopart
cveardies(FK)	Integer	No	Clave del catálogo cardies
cvetipoasocarea(FK)	Integer	No	Clave del catálogo ctipoasocarea
nomdiscespec	Varchar(255)	No	Nombre de la disciplina o especialidad

tbeneficiario : Tabla de beneficiarios de pólizas de participantes.

Nombre	Tipo de dato	Nulo	Descripción
idbenef(PK)	Integer	No	Identificador de la tabla tbeneficiario
idsegurotipopart(FK)	Integer	No	Identificador de la tabla tsegurotipopart
nombenef	Varchar(300)	Si	Nombre
parentesco	Varchar(100)	Si	Parentesco
porcentaje	Numeric(4,2)	Si	Porcentaje
fecnacbenef	Smalldatetime	Si	Fecha de nacimiento
fecantigpolbenef	Varchar(50)	Si	Fecha antigüedad en la póliza
observacion	Varchar(1000)	Si	Observación
estado	Char(1)	No	Estado de la tabla tbeneficiario

tbitacora : Tabla de la bitácora de cambios en los elementos del sistema.

Nombre	Tipo de dato	Nulo	Descripción
idbitacora(PK)	Integer	No	Identificador de la tabla tbitacora
idsitgestor(FK)	Integer	No	Identificador de la tabla tsitgestor
idtipopart(FK)	Integer	No	Identificador de la tabla ttipopart
fecmodifi	Datetime	Si	Fecha de modificación
nomforma	Varchar(255)	No	Nombre de la forma
nomelemento	Varchar(255)	No	Nombre del elemento
valorant	Varchar(1000)	No	Valor anterior
valornuevo	Varchar(1000)	No	Valor nuevo

tciclosoli : Tabla de los ciclos de solicitudes.

Nombre	Tipo de dato	Nulo	Descripción
idciclosoli(PK)	Integer	No	Identificador de la tabla tciclosoli
idciclosoliparent(FK)	Integer	No	Identificador de la tabla tciclosoli
cveetapaconvusun(FK)	Integer	No	Clave del catálogo cetapaconvusun

tcompoliza : Tabla que relaciona pólizas y compañías vigentes.

Nombre	Tipo de dato	Nulo	Descripción
idcompoliza(PK)	Integer	No	Identificador de la tabla tcompoliza
cvecompania(FK)	Integer	No	Clave del catálogo ccompanias
cvepoliza(FK)	Integer	No	Clave del catálogo cpolizas
Sumaaseg	Numeric(9,2)	Si	Suma asegurada
fecliivig	Smalldatetime	No	Fecha inicio de la vigencia
fecfinvig	Smalldatetime	No	Fecha fin de la vigencia
estado	Char(1)	No	Estado de la tabla tcompoliza

tctabancotipopart : Tabla de cuentas bancarias de participantes.

Nombre	Tipo de dato	Nulo	Descripción
idctabancotipopart(PK)	Integer	No	Identificador de la tabla tctabancotipopart
idtipopart(FK)	Integer	No	Identificador de la tabla ttipopart
cvecompania(FK)	Integer	Si	Clave del catálogo ccompanias
cvetipoestim(FK)	Integer	Si	Clave del catálogo ctipoestim
numsucursal	Varchar(100)	Si	Número de sucursal
numctabancaria	Varchar(50)	Si	Número de cuenta bancaria
monto	Numeric(8,2)	No	Monto

tdesigtipopart : Tabla de designaciones de emeritazgo.

Nombre	Tipo de dato	Nulo	Descripción
iddesigtipopart(PK)	Integer	No	Identificador de la tabla tdesigtipopart
idtipopart(FK)	Integer	No	Identificador de la tabla ttipopart
cvedesig(FK)	Integer	No	Clave del catálogo cdesignaciones
cvedepsubdepnom(FK)	Integer	No	Clave del catálogo cdepsubdep
cvedepsubdep(FK)	Integer	No	Clave del catálogo cdepsubdep
fecdesig	Smalldatetime	Si	Fecha de designación
cvecateg(FK)	Integer	No	Clave del catálogo ccategorias
estado	Char(1)	No	Estado de la tabla tdesigtipopart

tdisttipopart : Tabla de distinciones de participantes.

Nombre	Tipo de dato	Nulo	Descripción
iddisttipopart(PK)	Integer	No	Identificador de la tabla tdisttipopart
idtipopart(FK)	Integer	No	Identificador de la tabla ttipopart
cvedist(FK)	Integer	No	Clave del catálogo cdistinciones
cveambitodist(FK)	Integer	No	Clave del catálogo cambitodist
cveentregion(FK)	Integer	No	Clave del catálogo centregion
anio	Char(4)	Si	Año en que se recibe la distinción
cveinst(FK)	Integer	No	Clave del catálogo cinstituciones
otrains	Varchar(255)	No	Nombre de otra institución
observación	Varchar(1000)	No	Observaciones sobre distinciones
estado	Char(1)	No	Estado de la tabla tdisttipopart

tdomtipopart : Tabla que relaciona domicilios con participantes.

Nombre	Tipo de dato	Nulo	Descripción
iddomtipopart(PK)	Integer	No	Identificador de la tabla tdomtipopart
iddomrelasun(FK)	Integer	No	Identificador de la tabla tdomrelasun
idtipopart(FK)	Integer	No	Identificador de la tabla ttipopart

tedociviltipopart : Tabla que relaciona el estado civil con participantes.

Nombre	Tipo de dato	Nulo	Descripción
idedociviltipopart(PK)	Integer	No	Identificador de la tabla tedociviltipopart
idedocivilasun(FK)	Integer	No	Identificador de la tabla tedocivilasun
idtipopart(FK)	Integer	No	Identificador de la tabla ttipopart

tedosaludtipopart : Tabla del estado de salud de participantes.

Nombre	Tipo de dato	Nulo	Descripción
idedosaludtipopart(PK)	Integer	No	Identificador de la tabla tedosaludtipopart
idtipopart(FK)	Integer	No	Identificador de la tabla ttipopart
cveedosalud(FK)	Integer	No	Clave del catálogo cedosalud
fecedosalud	Smalldatetime	Si	Fecha del estado de salud
observación	Varchar(1000)	Si	Observaciones
estado	Char(1)	No	Estado de la tabla tedosaludtipopart

telemdiftipopart : Tabla de elementos de difusión de participantes.

Nombre	Tipo de dato	Nulo	Descripción
idelemdiftipopart(PK)	Integer	No	Identificador de la tabla telemdiftipopart
idtipopart(FK)	Integer	No	Identificador de la tabla ttipopart
nomelem	Varchar(255)	No	Nombre del elemento
cvetipoelemdif(FK)	Integer	No	Clave del catálogo ctipoelemdif
cvetiporoldif(FK)	Integer	No	Clave del catálogo ctiporoldif
cveentregion(FK)	Integer	Si	Clave del catálogo centregion
lugardif	Varchar(255)	Si	Lugar
fecdif	Smalldatetime	Si	Fecha
observación	Varchar(1000)	Si	Observación
estado	Char(1)	No	Estado de la tabla telemdiftipopart

tgestor : Tabla de control de los gestores del sistema.

Nombre	Tipo de dato	Nulo	Descripción
idgestor(PK)	Integer	No	Identificador de la tabla tgestor
idrolper(FK)	Integer	No	Identificador de la tabla trolper
cvetipogestor(FK)	Integer	No	Clave del catálogo ctipogestor
cvedepsubdep(FK)	Integer	No	Clave del catálogo cdepsubdep

tlugarnactipopart : Tabla del lugar de nacimiento de participantes.

Nombre	Tipo de dato	Nulo	Descripción
idlugnactipopart(PK)	Integer	No	Identificador de la tabla tlugnactipopart
idtipopart(FK)	Integer	No	Identificador de la tabla ttipopart
cveentregion(FK)	Integer	No	Clave del catálogo centregion
lugnac	Varchar(255)	Si	Lugar de nacimiento

tmovpadeci : Tabla de movimientos por padecimiento de participantes.

Nombre	Tipo de dato	Nulo	Descripción
idmovpadeci(PK)	Integer	No	Identificador de la tabla tmovpadeci
idpadecitipopart(FK)	Integer	No	Identificador de la tabla tpadecitipopart
idsegurotipopart(FK)	Integer	No	Identificador de la tabla tsegurotipopart
movimiento	Varchar(1000)	No	Movimiento
numreclamo	Varchar(100)	Si	Número de reclamación
cvetipopagomovpadeci(FK)	Integer	No	Clave del catálogo ctipopagomovpadeci
fecpresent	Varchar(50)	Si	Fecha de presentación
fecdeposito	Smalldatetime	No	Fecha de depósito
montoreclama	Numeric(9,2)	Si	Monto reclamado
montonocubre	Numeric(9,2)	Si	Monto que no cubre
montodeduce	Numeric(9,2)	Si	Monto deducible
montocoaseguro	Numeric(9,2)	Si	Monto de coaseguro
observación	Varchar(1000)	Si	Observación
estado	Char(1)	No	Estado de la tabla tmovpadeci

tnivacadtipopart : Tabla que relaciona niveles académicos con participantes.

Nombre	Tipo de dato	Nulo	Descripción
idnivacadtipopart(PK)	Integer	No	Identificador de la tabla tnivacadtipopart
idtipopart(FK)	Integer	No	Identificador de la tabla ttipopart
idnivacadrelasun(FK)	Integer	No	Identificador de la tabla tnivacadrelasun
titulo	Varchar(255)	No	Título
abrevtit	Varchar(100)	Si	Abreviatura del título
maxnivacad	Char(1)	Si	Máximo nivel académico
aniotit	Char(4)	Si	Año de titulación
otrains	Varchar(255)	Si	Institución
cveentregion(FK)	Integer	Si	Clave del catálogo centregion
observación	Varchar(1000)	Si	Observación
estado	Char(1)	No	Estado de la tabla tnivacadtipopart

tpadecitipart : Tabla para almacenar los padecimientos de participantes.

Nombre	Tipo de dato	Nulo	Descripción
idpadecitipart(PK)	Integer	No	Identificador de la tabla tpadecitipart
idtipopart(FK)	Integer	No	Identificador de la tabla ttipopart
nompadeci	Varchar(1000)	Si	Nombre
sumaini	Numeric(9,2)	No	Suma inicial
fecinipadeci	Varchar(50)	Si	Nombre
estado	Char(1)	No	Estado de la tabla tpadecitipart

tsegurotipopart : Tabla de pólizas de seguros de participantes.

Nombre	Tipo de dato	Nulo	Descripción
idsegurotipopart(PK)	Integer	No	Identificador de la tabla tnivacadtipopart
idtipopart(FK)	Integer	No	Identificador de la tabla ttipopart
idcompoliza(FK)	Integer	No	Identificador de la tabla tcompoliza
fecpresent	Smalldatetime	Si	Fecha de presentación
fecrecepdep	Smalldatetime	Si	Fecha de recepción de la dependencia
fecrecep	Smalldatetime	Si	Fecha de recepción
numpoliza	Varchar(50)	Si	Número de póliza
fecantigtit	Varchar(50)	Si	Fecha de antigüedad del titular
deducible	Numeric(9,2)	Si	Monto deducible
coaseguro	Numeric(9,2)	Si	Monto coaseguro
observación	Varchar(1000)	Si	Observación
estado	Char(1)	No	Estado de la tabla tsegurotipopart

tsitciclosoli : Tabla de situaciones de la solicitud PERPAE.

Nombre	Tipo de dato	Nulo	Descripción
idsitciclosoli(PK)	Integer	No	Identificador de la tabla tsitciclosoli
idciclosoli(FK)	Integer	No	Identificador de la tabla tciclosoli
cvesitciclosoli(FK)	Integer	No	Clave del catálogo csitciclosoli
fecregsitciclosoli	Smalldatetime	Si	Fecha de registro
estado	Char(1)	No	Estado de la tabla tsitciclosoli

tsitgestor : Tabla de situaciones de gestores en el sistema.

Nombre	Tipo de dato	Nulo	Descripción
idsitgestor(PK)	Integer	No	Identificador de la tabla tsitgestor
idgestor(FK)	Integer	No	Identificador de la tabla tgestor
cvesitgestor(FK)	Integer	No	Clave del catálogo csitgestor
fecregsitgestor	Smalldatetime	No	Fecha de registro
estado	Char(1)	No	Estado de la tabla tsitgestor

tsitlaboraltipopart : Tabla de situación laboral de participantes.

Nombre	Tipo de dato	Nulo	Descripción
idsitlaboraltipopart(PK)	Integer	No	Identificador de la tabla ttipopart
idtipopart(FK)	Integer	No	Identificador de la tabla ttipopart
antigüedad	Integer	Si	Antigüedad
fecingunam	Varchar(50)	Si	Fecha de ingreso a la UNAM
cveleyissste(FK)	Integer	No	Clave del catálogo cleyissste
hrsasig	Numeric(4,1)	Si	Número de horas de asignatura
fecjub	Varchar(50)	Si	Fecha de jubilación
cvetipohrsasigjub(FK)	Integer	Si	Clave del catálogo ctipohrsasig
cvetipohrsasig(FK)	Integer	Si	Clave del catálogo ctipohrsasig
hrsasigjub	Numeric(4,1)	Si	Número de horas de asignatura de jubilación
cvecateg(FK)	Integer	Si	Clave del catálogo ccategorias
cvedepsubdep(FK)	Integer	Si	Clave del catálogo cdepsubdep
jubactivo	Char(1)	Si	Jubilado activo

tsittipart : Tabla de situaciones de participantes.

Nombre	Tipo de dato	Nulo	Descripción
idsittipart(PK)	Integer	No	Identificador de la tabla tsittipart
idtipopart(FK)	Integer	No	Identificador de la tabla ttipopart
cvesittipart(FK)	Integer	No	Clave del catálogo csittipart
fecregsittipart	Smalldatetime	No	Fecha de registro
estado	Char(1)	No	Estado de la tabla tsittipart

tteltipopart : Tabla que relaciona teléfonos con participantes.

Nombre	Tipo de dato	Nulo	Descripción
idteltipopart(PK)	Integer	No	Identificador de la tabla tteltipopart
idtelrelasun(FK)	Integer	No	Identificador de la tabla ttelrelasun
idtipopart(FK)	Integer	No	Identificador de la tabla ttipopart

ttipoobstipopart : Tabla que relaciona tipos de observaciones con tipos de participación.

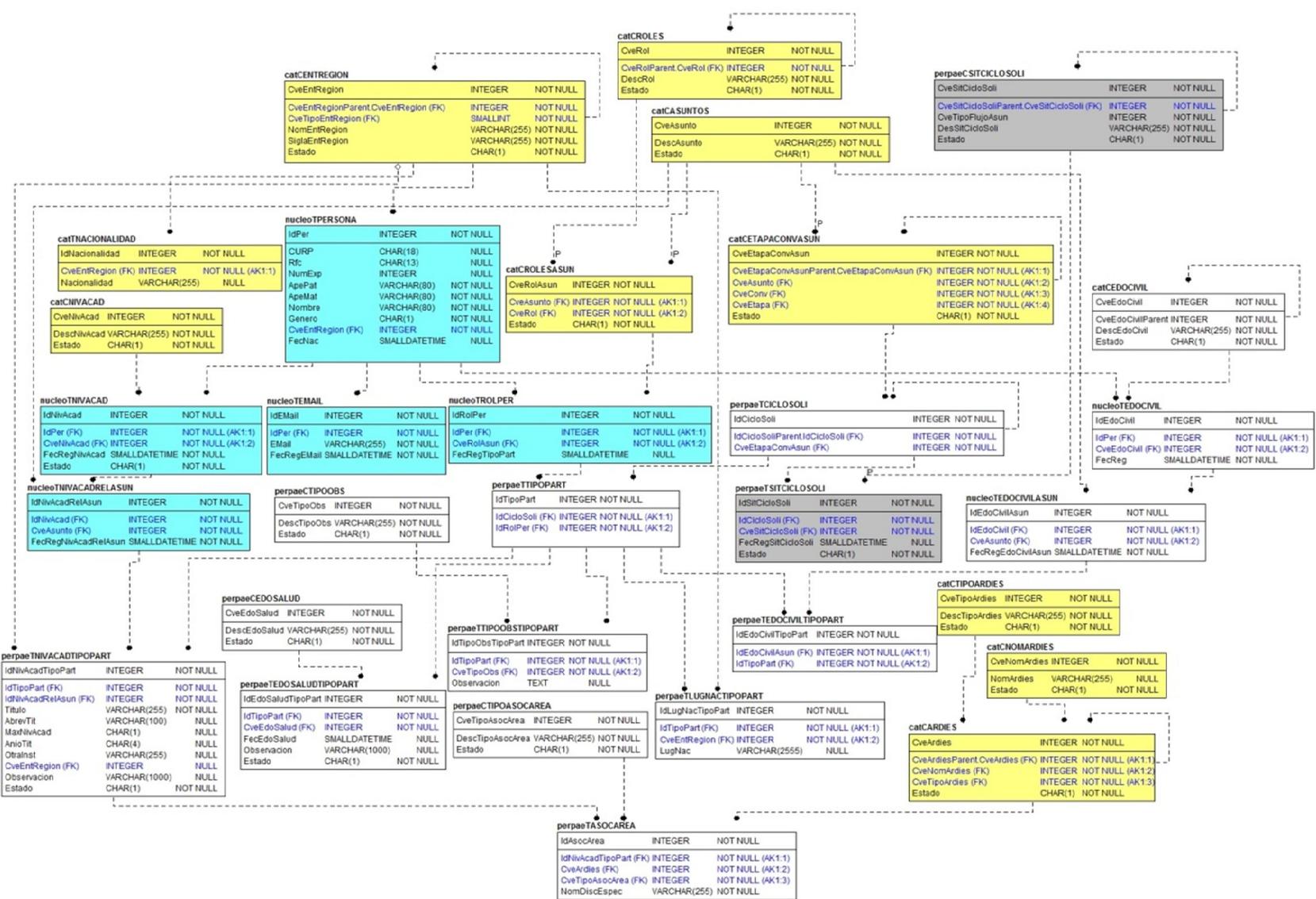
Nombre	Tipo de dato	Nulo	Descripción
idtipoobstipopart(PK)	Integer	No	Identificador de la tabla ttipoobstipopart
idtipopart(FK)	Integer	No	Identificador de la tabla ttipopart
cvetipoobs(FK)	Integer	No	Clave del catálogo ctipoobs
observación	Text	Si	Observación

ttipopart : Tabla de tipos de participaciones en el PERPAE.

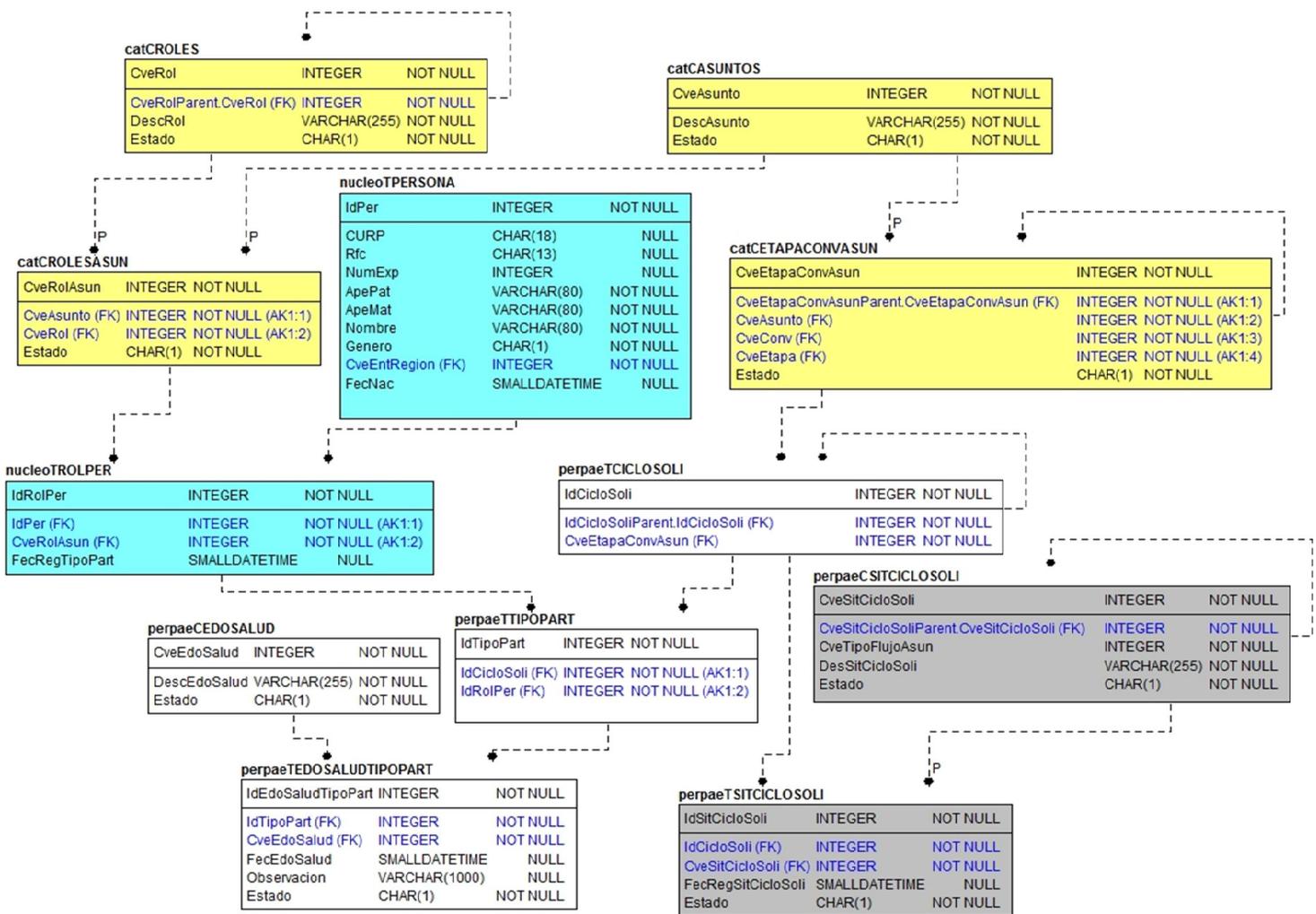
Nombre	Tipo de dato	Nulo	Descripción
idtipopart(PK)	Integer	No	Identificador de la tabla ttipopart
idciclosoli(FK)	Integer	No	Identificador de la tabla tciclosoli
idrolper(FK)	Integer	No	Identificador de la tabla trolper

MODELO DE DATOS

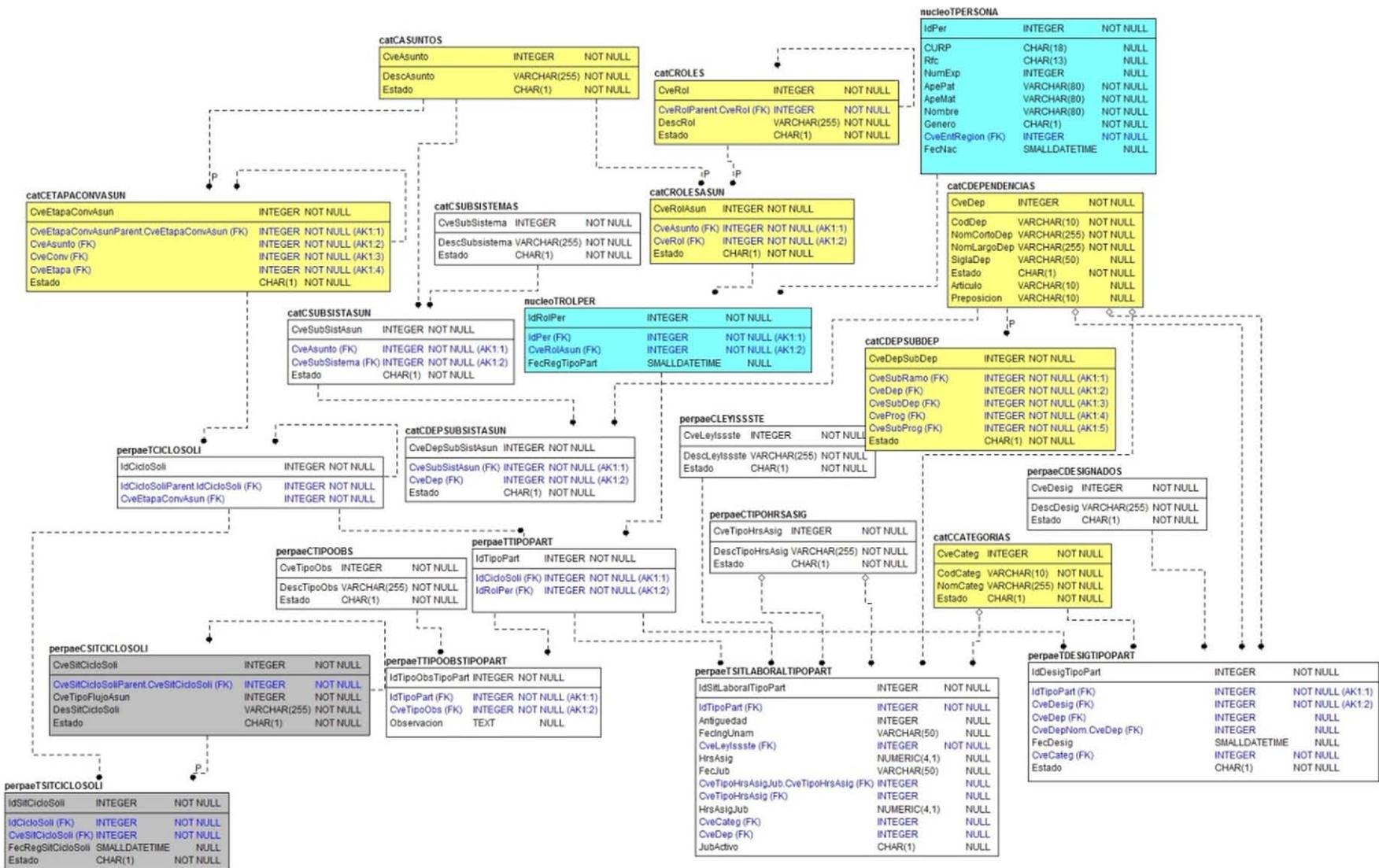
Modelo "Datos personales"



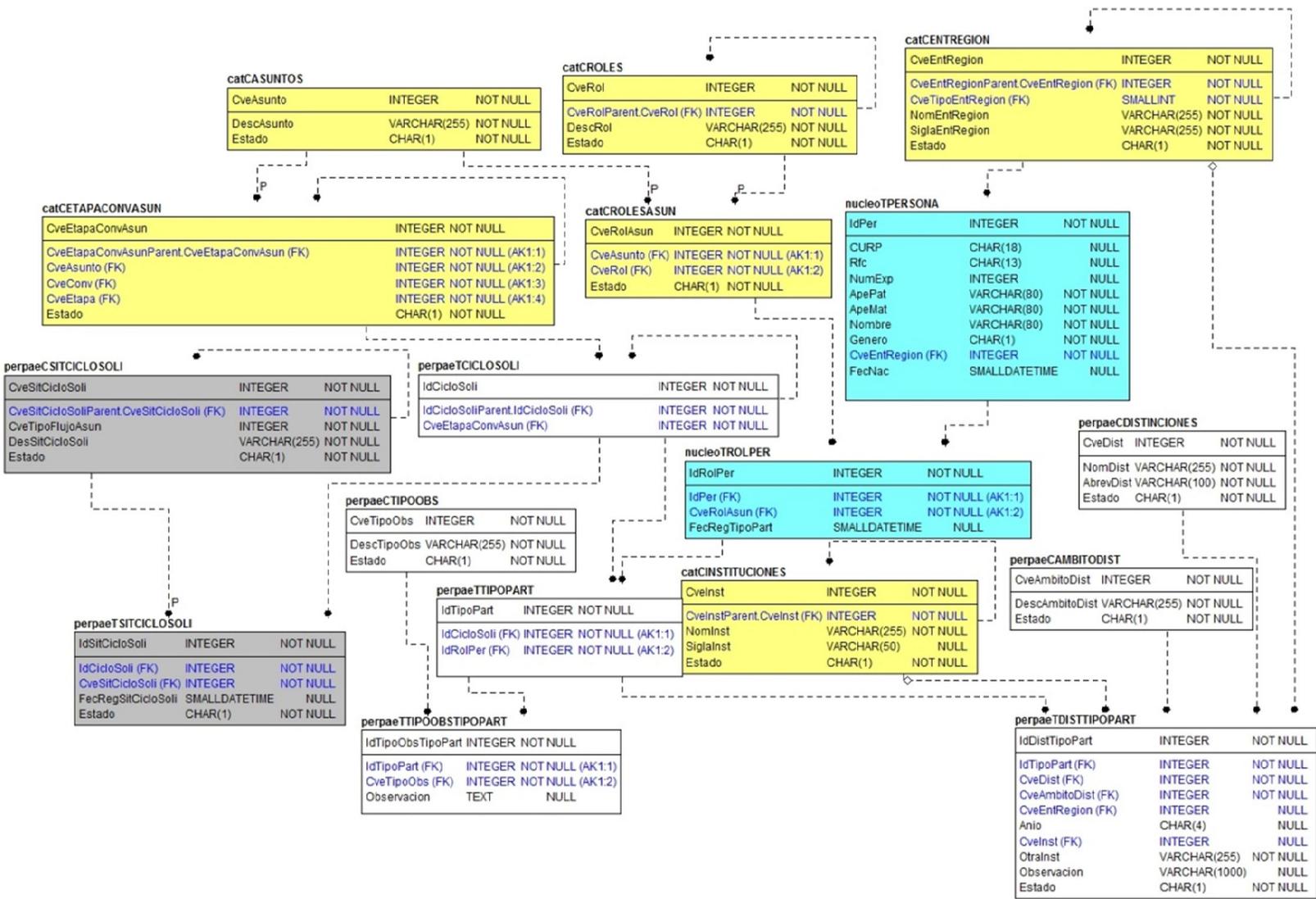
Modelo "Situación personal"



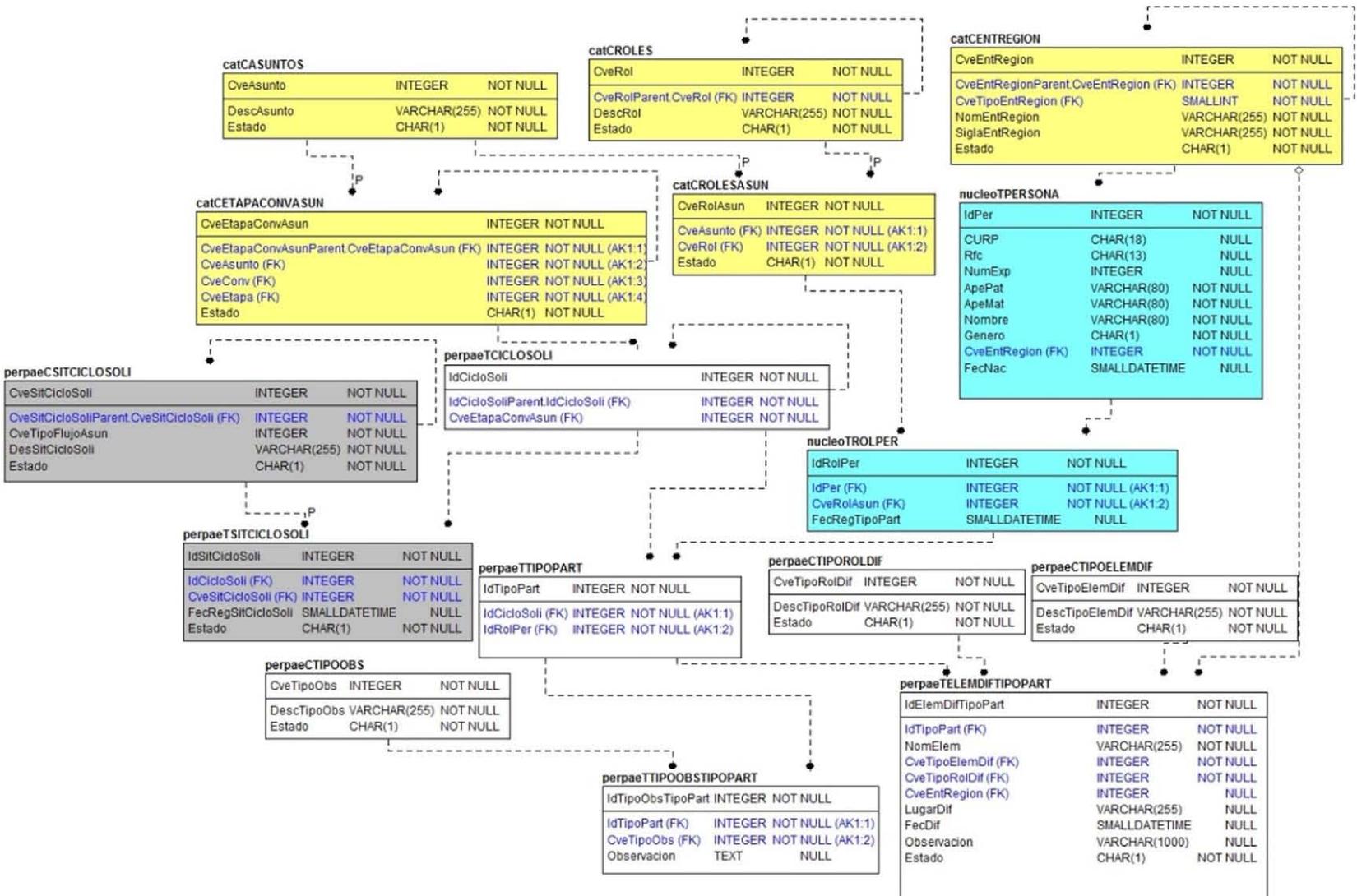
Modelo "Situación laboral y designaciones"



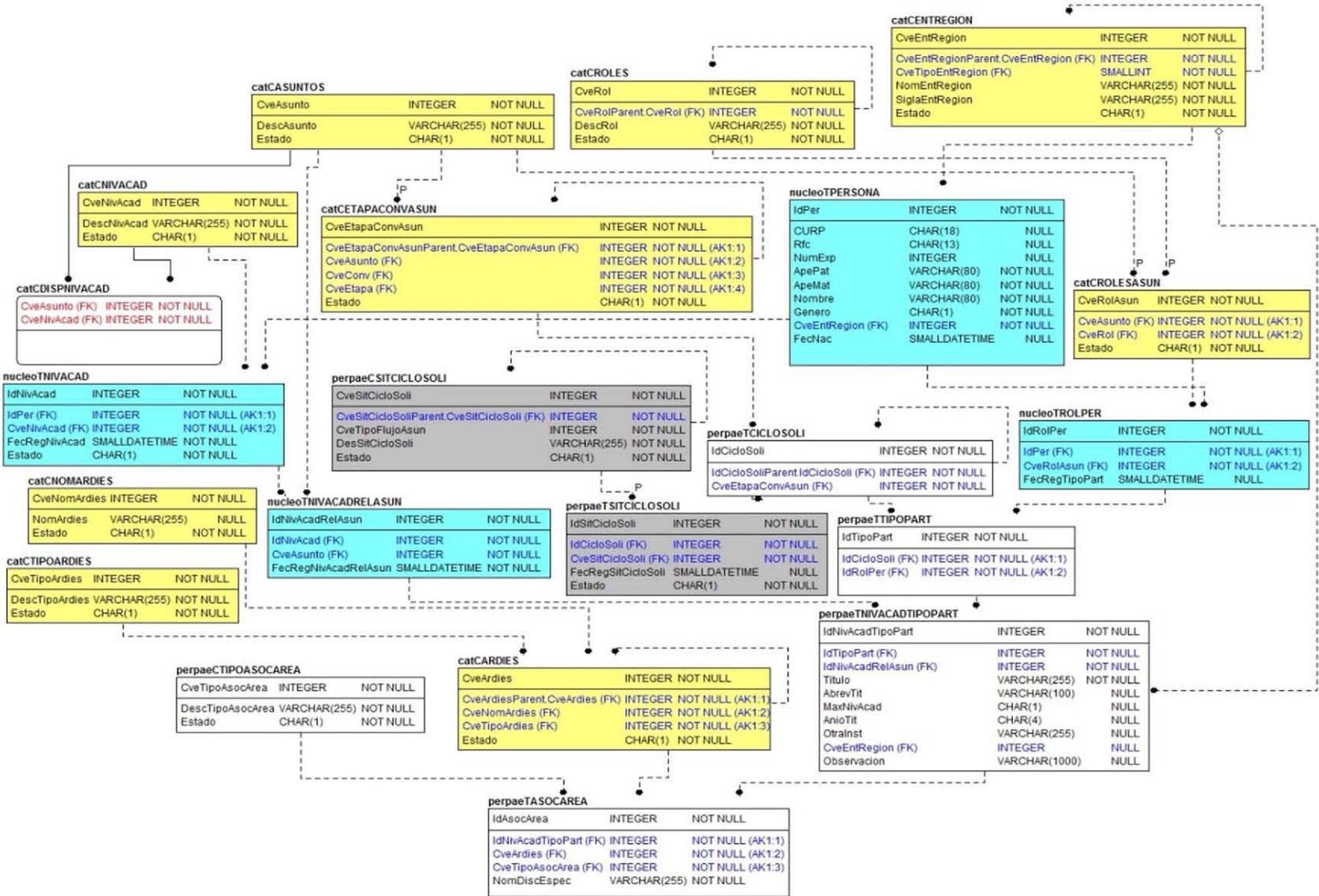
Modelo "Distinciones"



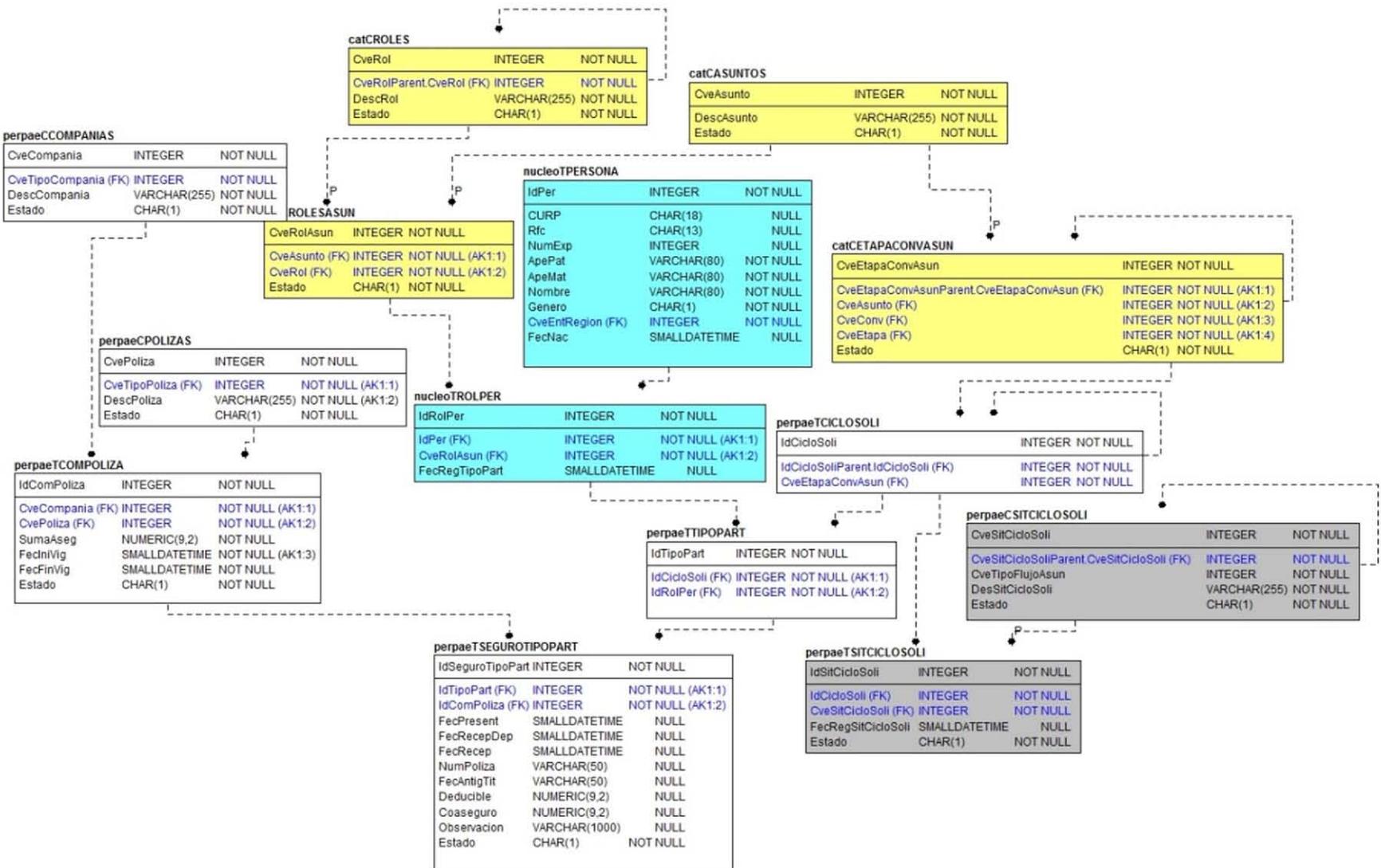
Modelo "Actividades de difusión"



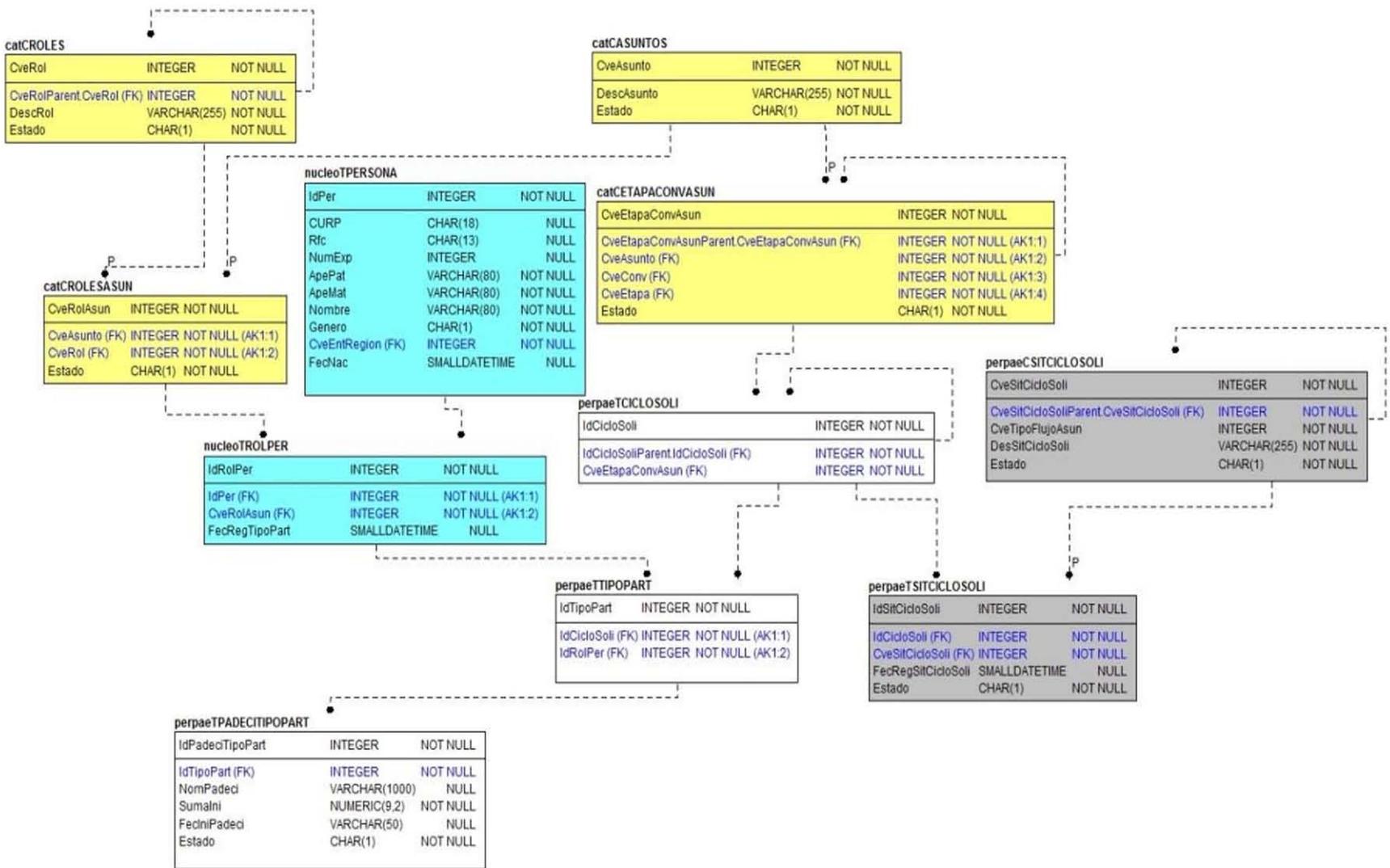
Modelo "Estudios"



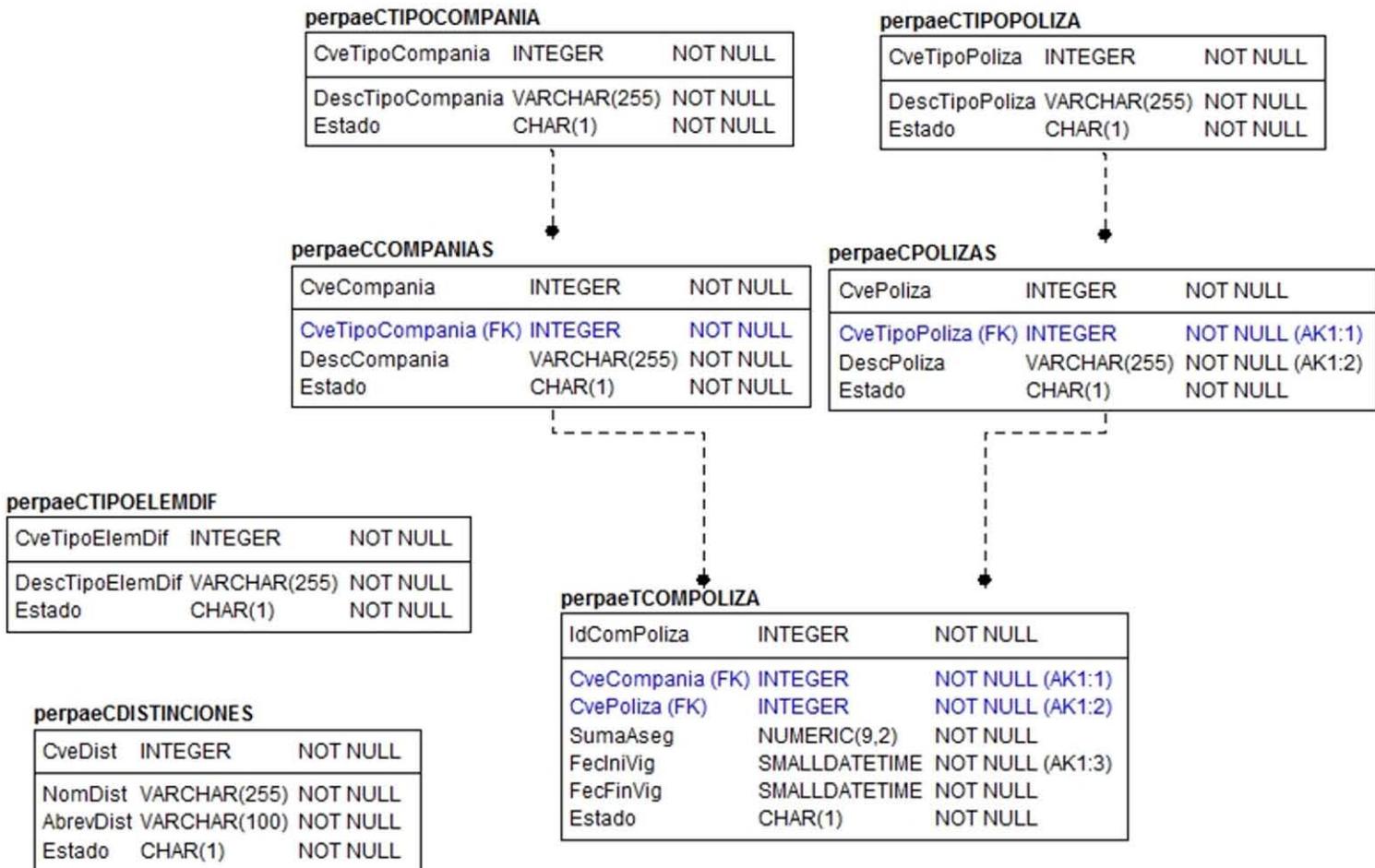
Modelo "Pólizas de seguros"



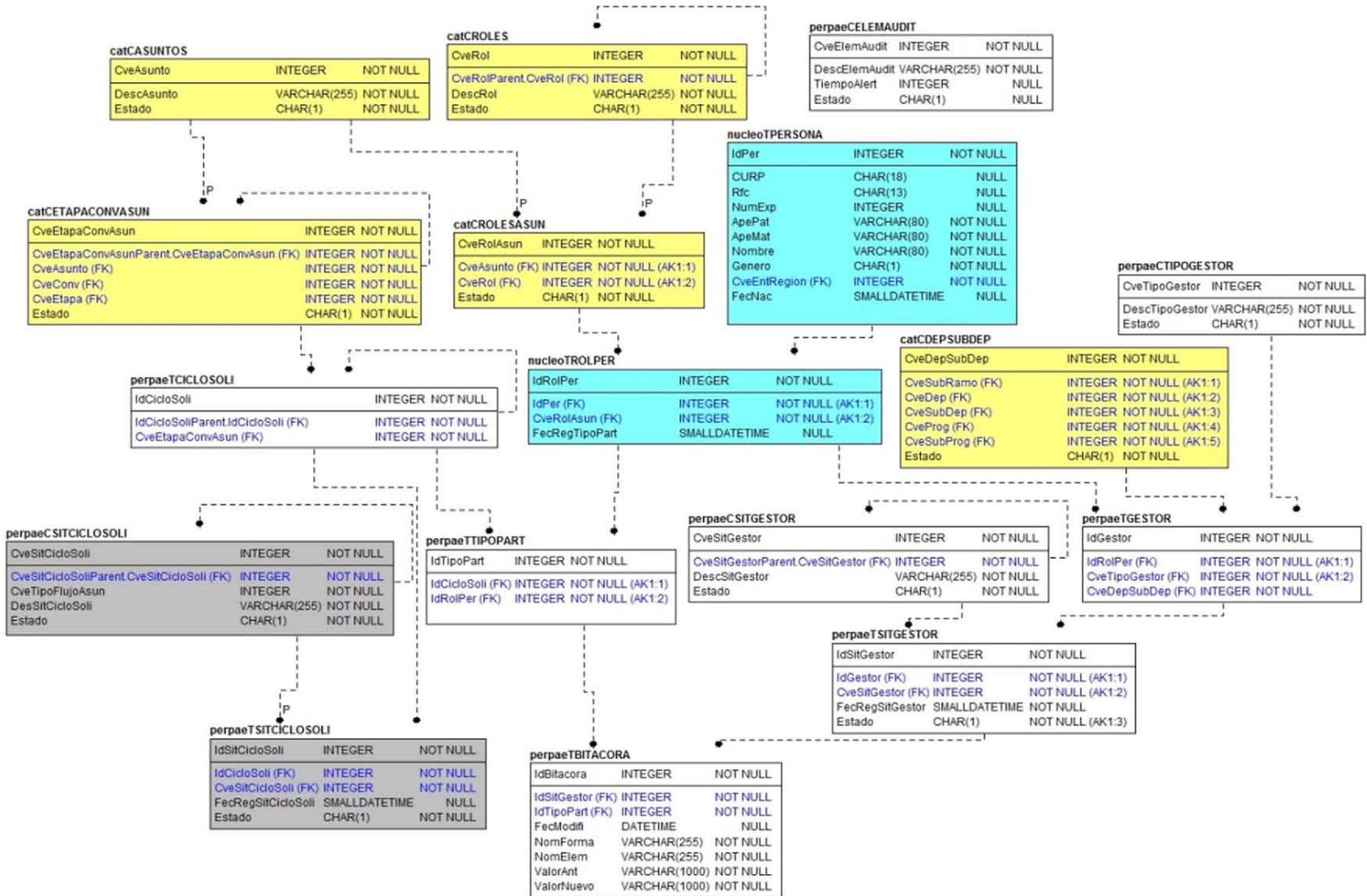
Modelo "Padecimientos"



Modelo "Catálogos del sistema"



Modelo "Alertas generadas por el sistema"



INTERFACES GRÁFICAS

Página “Inicio”

INICIO

Bienvenido al sistema PERPAE

Alerta: Los siguientes eméritos tienen información que no ha sido actualizada recientemente
(Para mayor detalle, consultar la pantalla de alertas)

Núm. Exp.	Nombre del emérito
7088	NOMBRE DE EMÉRITO
10930	NOMBRE DE EMÉRITO
34029	NOMBRE DE EMÉRITO
38132	NOMBRE DE EMÉRITO
40071	NOMBRE DE EMÉRITO
47636	NOMBRE DE EMÉRITO

Alerta: Las siguientes pólizas están próximas a expirar
(Para mayor detalle, consultar la pantalla de pólizas contratadas)

Póliza	Tipo	Compañía	Fecha inicio de vigencia	Fecha fin de vigencia
póliza vida 1	De vida	001	6/08/2012	2/12/2012

Fig. 1 - Página “Inicio”

Página “Eméritos”

EMÉRITOS

RFC

Núm. de exp.

Apellido paterno

REGISTROS DE NÓMINA

Núm. de exp.	RFC	Nombre completo	Género	Antigüedad
12205	SESL2804188D0	ACADÉMICO DE LA UNAM	F	52

REGISTROS DE PERPAE

Núm. de exp.	RFC	Nombre completo	Género	Antigüedad	Fecha de Ingreso	Situación	Eliminar
12205	SESL2804188D0	NOMBRE DE EMÉRITO	F	65	08/08/1999	Jubilado activo	<input type="checkbox"/>

Fig. 2 - Página “Eméritos”

Página “Datos personales”

DATOS PERSONALES

Nombre: NOMBRE DE EMÉRITO		Situación: Jubilado activo	
RFC: SESL280418BD0		Núm. Exp: 12205	
Género: F		Edad: 84 años 3 meses al 27/07/2012	

RFC	Homoclave Hda.	Núm. Exp.
<input type="text" value="SESL280418"/>	<input type="text" value="BD0"/>	<input type="text" value="12205"/>

Nombre(s)

Apellido paterno	Apellido materno
<input type="text"/>	<input type="text"/>

*Nombre completo que se utilizará en el PERPAE

CURP	*Estado civil	Lugar de nacimiento
<input type="text"/>	<input type="text" value="Soltero"/>	<input type="text" value="Mexico DF"/>

*País de nacimiento	*Nacionalidad	Fecha de nacimiento
<input type="text" value="Islas Christmas"/>	<input type="text" value="Mexicana"/>	<input type="text" value="18/04/1928"/>

Género	Correo electrónico
<input type="text" value="F"/>	<input type="text"/>

Cónyuge:

Nombre		
<input type="text"/>		
Grado	Título	
<input type="text"/>	<input type="text"/>	
Fecha de nacimiento	Situación	Fecha de fallecimiento
<input type="text"/>	<input type="text"/>	<input type="text"/>

Ir a datos de cónyuge

Observaciones

* : Campos obligatorios

Fig. 3 - Página “Datos personales”

Página “Situación personal”

SITUACIÓN PERSONAL

Nombre: NOMBRE DE EMÉRITO RFC: SESL280418BD0 Género: F	Situación: Jubilado activo Núm. Exp: 12205 Edad: 84 años 3 meses al 27/07/2012
--	--

HISTORIAL DE ESTADO

Estado	Fecha	Observaciones	Eliminar
Regular	23/07/2012		<input type="checkbox"/>

ESTADO

*Estado actual

*Fecha de estado

Observaciones del estado

* : Campos obligatorios

Fig. 4 - Página “Situación personal”

Página “Domicilios”

DOMICILIO

Nombre: NOMBRE DE EMÉRITO RFC: SESL280418BD0 Género: F	Situación: Jubilado activo Núm. Exp: 12205 Edad: 84 años 3 meses al 27/07/2012
--	--

LISTA DE DOMICILIOS

Tipo de domicilio	Domicilio	Eliminar
Particular	Violeta, Col. San Jeronimo, C.P. 13205, México	<input type="checkbox"/>

DOMICILIO

*Tipo de domicilio

*Calle

*Colonia

Delegación

*Ciudad

*C.P.

*País

*Teléfono(s)

Fax

Observaciones generales de domicilios

* : Campos obligatorios

Fig. 5 - Página “Domicilios”

Página “Contactos”

CONTACTOS

Nombre: NOMBRE DE EMÉRITO	Situación: Jubilado activo
RFC: SESL280418BD0	Núm. Exp: 12205
Género: F	Edad: 84 años 3 meses al 27/07/2012

LISTA DE CONTACTOS

Tipo de contacto	Nombre	Eliminar
Asistente del Emérito	JUANA PEREZ PEREZ	<input type="checkbox"/>
Discípulo del Emérito	LAURA ADRIANA PEREZ GOMEZ	<input type="checkbox"/>

CONTACTO

*Nombre	*Apellido Paterno	*Apellido Materno
<input type="text"/>	<input type="text"/>	<input type="text"/>
*Tipo de contacto	RFC	*Género
<input type="text"/>	<input type="text"/>	<input type="text"/>
Grado	Título	Abrev. Título
<input type="text"/>	<input type="text"/>	<input type="text"/>
Fecha de nacimiento	Fecha de fallecimiento	
<input type="text"/>	<input type="text"/>	

DOMICILIO DEL CONTACTO

*Tipo de Domicilio		
<input type="text"/>		
*Calle	*Colonia	Delegación
<input type="text"/>	<input type="text"/>	<input type="text"/>
*Ciudad	*País	*C.P.
<input type="text"/>	<input type="text"/>	<input type="text"/>
*Teléfono(s)	Fax	Correo electrónico
<input type="text"/>	<input type="text"/>	<input type="text"/>

Observaciones generales de contactos

* : Campos obligatorios

Fig. 6 - Página “Contactos”

Página “Situación laboral”

SITUACIÓN LABORAL

Nombre: NOMBRE DE EMÉRITO RFC: SESL280418BD0 Género: F	Situación: Jubilado activo Núm. Exp: 12205 Edad: 84 años 3 meses al 27/07/2012
--	--

LISTA DE DESIGNACIONES

Designado como	Dependencia	Eliminar
Profesor emerito	Casa del Lago	<input type="checkbox"/>
Investigador emerito	Auditoría Interna	<input type="checkbox"/>

DESIGNACIÓN

*Designado como	*Dependencia
<input type="text"/>	<input type="text"/>
Subsistema	*Fecha designación
<input type="text"/>	<input type="text"/>
*Nombramiento	*Dependencia de adscripción
<input type="text"/>	<input type="text"/>

DATOS GENERALES DE SITUACIÓN LABORAL

*Antigüedad	*Fecha de ingreso a la UNAM	*Ley del ISSSTE
<input type="text" value="65"/>	<input type="text" value="13/07/2012"/>	<input type="text" value="Artículo 10 Transitorio"/>
Tipo de horas de asignatura	Horas de asignatura	
<input type="text" value="A"/>	<input type="text"/>	

DATOS DE JUBILACIÓN

Fecha de jubilación	Nombramiento	Dependencia
<input type="text" value="may 2011"/>	<input type="text" value="PROF ASIG A"/>	<input type="text" value="Coord. y Cons. Tec. de Humanidad"/>
Situación	Tipo de horas de asignatura	Horas de asignatura
<input type="text" value="Activo"/>	<input type="text" value="B"/>	<input type="text"/>

Observaciones generales

* : Campos obligatorios

Fig. 7 - Página “Situación laboral”

Página “Distinciones”

DISTINCIONES

Nombre: NOMBRE DE EMÉRITO RFC: SESL280418BD0 Género: F	Situación: Jubilado activo Núm. Exp: 12205 Edad al 12/10/2012: 84 años 5 meses
--	--

LISTA DE DISTINCIONES

Distinción	Abreviatura	Año	Eliminar
Premio Universidad Nacional	PUN		<input type="checkbox"/>
Sistema Nacional de Investigadores	SNI	1995	<input type="checkbox"/>

DISTINCIÓN

*Distinción	Año
<input type="text"/>	<input type="text"/>
*País	*Ámbito
<input type="text"/>	<input type="text"/>
Otorga la UNAM	Institución que otorga
<input type="text"/>	<input type="text"/>
Observaciones de la distinción	
<input type="text"/>	

* : Campos obligatorios

Fig. 8 - Página “Distinciones”

Página “Difusión”

DIFUSIÓN

Nombre: NOMBRE DE EMÉRITO RFC: SESL280418BD0 Género: F	Situación: Jubilado activo Núm. Exp: 12205 Edad: 84 años 3 meses al 27/07/2012
--	--

LISTA DE DIFUSIONES

Elemento	Tipo de elemento	Tipo de participación	Eliminar
Conferencia universitaria	CONFERENCIA	Primera persona	<input type="checkbox"/>

DIFUSIÓN

*Tipo de elemento	*Elemento
<input type="text"/>	<input type="text"/>
*Tipo de participación	Fecha
<input type="text"/>	<input type="text"/>
*País	Lugar
<input type="text"/>	<input type="text"/>
Observaciones de la difusión	
<input type="text"/>	

* : Campos obligatorios

Fig. 9 - Página “Difusión”

Página “Estudios”

ESTUDIOS

Nombre: NOMBRE DE EMÉRITO	Situación: Jubilado activo
RFC: SESL280418BDO	Núm. Exp: 12205
Género: F	Edad: 84 años 3 meses al 27/07/2012

LISTA DE ESTUDIOS

Grado	Título	Área de disciplina	Máximo nivel	Eliminar
Maestría	Maestría en derecho	III.Ciencias sociales	No	<input type="checkbox"/>
Doctorado	Doctorado en Economía	III.Ciencias sociales	SI	<input type="checkbox"/>

ESTUDIO

*Grado

*Abrev. Título

*Pais

Estudio realizado en la UNAM

*Área de disciplina

*Especialidad

*Título

Año de titulación

Máximo nivel [Seleccione si es el máximo nivel para PERPAE]

Institución donde realizó el estudio

*Área de disciplina

*Área de especialidad

Observaciones del estudio

* : Campos obligatorios

Fig. 10 - Página “Estudios”

Página “Cuenta bancaria”

CUENTA BANCARIA

Nombre: NOMBRE DE EMÉRITO	Situación: Jubilado activo
RFC: SESL280418BDO	Núm. Exp: 12205
Género: F	Edad al 12/10/2012: 84 años 5 meses

Nombre del banco

*Tipo de pago

*Monto

Número de sucursal

Número de cuenta

Fecha de última actualización

Observaciones

* : Campos obligatorios

Fig. 11 - Página “Cuenta bancaria”

Página “Pólizas”

PÓLIZAS

Nombre: NOMBRE DE EMÉRITO	Situación: Jubilado activo
RFC: SESL280418BDD0	Núm. Exp: 12205
Género: F	Edad: 84 años 3 meses al 27/07/2012

LISTA DE PÓLIZAS DEL EMÉRITO (VIGENTES)

Ver vigentes [Ver histórico](#)

Póliza	Tipo	Compañía	Suma	Fecha inicio	Fecha fin	Eliminar
Poliza de vida grupal UNAM	De vida	Inbursa	1100000.0	2/06/2012	3/06/2014	<input type="checkbox"/>

*Póliza

* : Campos obligatorios

Fig. 12 - Página “Pólizas”

Página “Póliza de vida”

PÓLIZA DE VIDA

Nombre: NOMBRE DE EMÉRITO	Situación: Jubilado activo
RFC: SESL280418BDD0	Núm. Exp: 12205
Género: F	Edad al 12/10/2012: 84 años 5 meses

Póliza: Poliza de vida grupal UNAM Compañía: Inbursa
 Suma: 1100000.0 Vigencia: del 02/06/2012 al 03/06/2014

BENEFICIARIO

*Nombre del beneficiario

*Parentesco *Porcentaje

Observaciones

DATOS GENERALES DE LA PÓLIZA

Fecha de presentación Fecha de recepción

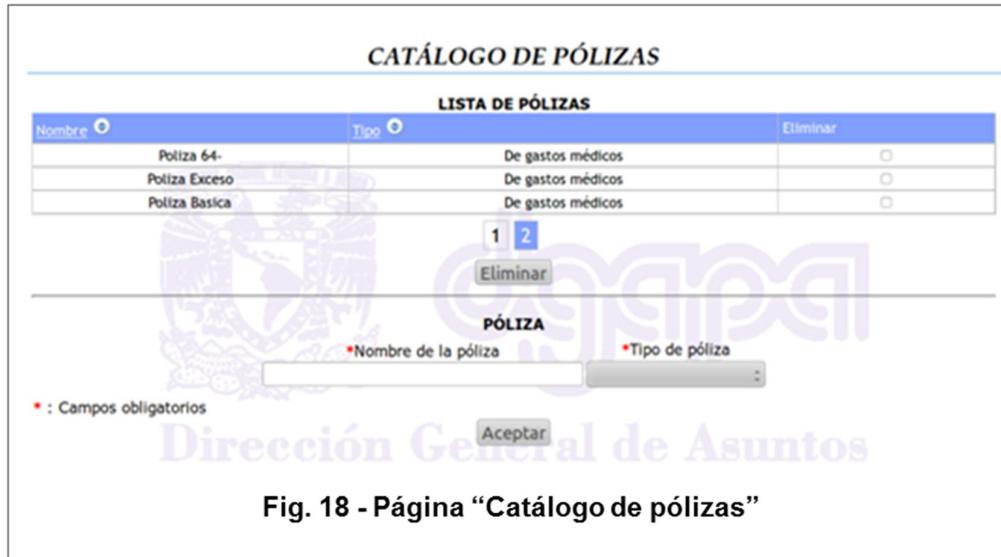
Fecha de recepción de dependencia Fecha últ. actualización

Observaciones

* : Campos obligatorios

Fig. 13 - Página “Póliza de vida”

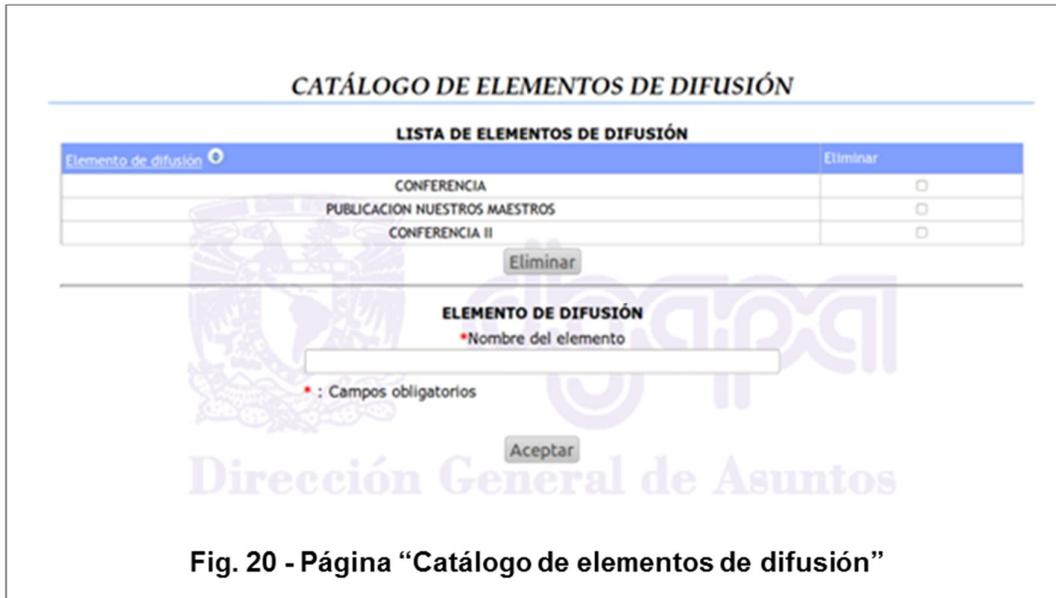
Página “Catálogo de pólizas”



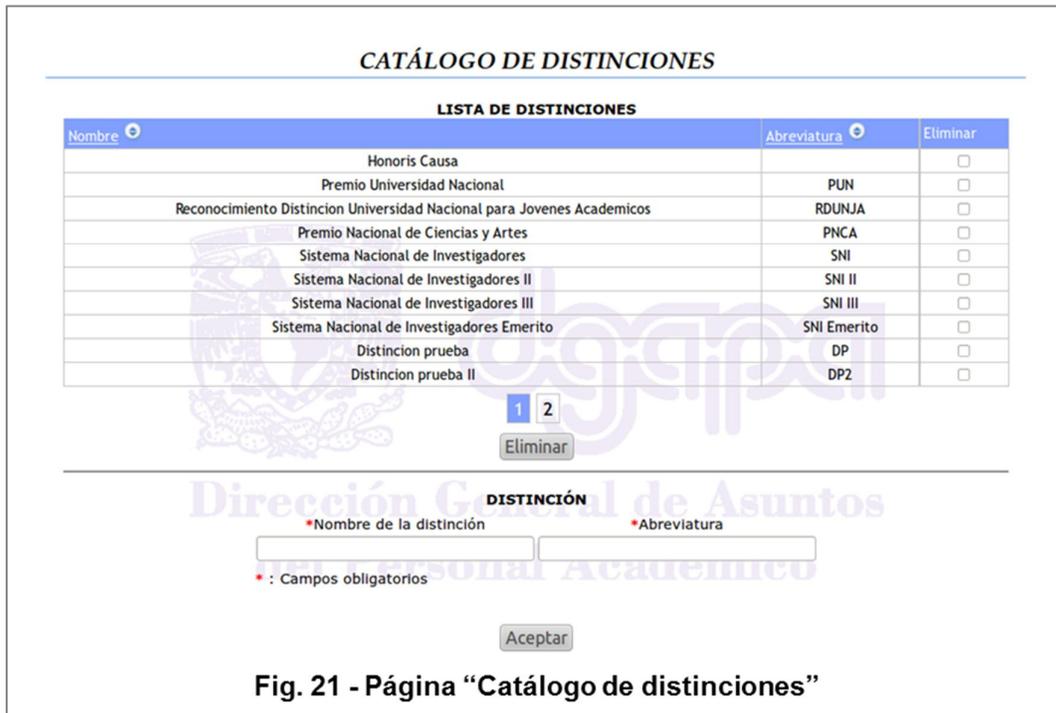
Página “Catálogo de compañías”



Página “Catálogo de elementos de difusión”



Página “Catálogo de distinciones”



Página “Pólizas contratadas”

PÓLIZAS CONTRATADAS

LISTA DE PÓLIZAS CONTRATADAS (VIGENTES)

Ver vigentes [Ver histórico](#)

Póliza	Tipo	Compañía	Suma asegurada	Fecha inicio	Fecha fin	Eliminar
Poliza 25+	De gastos médicos	ING	2000.0	8/09/2012	7/09/2014	<input type="checkbox"/>
Poliza 64-	De gastos médicos	Banco Azteca	1000000.0	20/09/2012	29/09/2012	<input type="checkbox"/>
Poliza 64-	De gastos médicos	ING	50000.0	12/09/2012	8/09/2014	<input type="checkbox"/>
Poliza 64-	De gastos médicos	Inbursa	50000.0	7/09/2012	16/09/2013	<input type="checkbox"/>
Poliza Basica	De gastos médicos	Inbursa	1000.0	12/06/2012	13/12/2012	<input type="checkbox"/>
Poliza Exceso	De gastos médicos	Inbursa	2000000.0	3/08/2012	18/08/2015	<input type="checkbox"/>
Poliza de vida grupal UNAM	De vida	Inbursa	1100000.0	2/06/2012	3/06/2014	<input type="checkbox"/>
Poliza de vida grupal UNAM	De vida	ING	1100000.0	13/06/2012	27/09/2012	<input type="checkbox"/>
Poliza de vida grupal UNAM	De vida	Grupo Nacional Provincial (GNP)	300000.0	1/09/2012	6/09/2015	<input type="checkbox"/>
poliza vida 1	De vida	001	1100000.0	6/08/2012	2/12/2012	<input type="checkbox"/>

1 2
Eliminar

COMPAÑÍA - PÓLIZA

*Nombre de la Compañía: *Póliza: *Suma de la póliza:

*Fecha inicio vigencia: *Fecha fin vigencia:

* : Campos obligatorios

Aceptar

Fig. 22 - Página “Pólizas contratadas”

Página “Alertas”

ALERTAS

Eméritos con información no actualizada recientemente

RFC: Apellido paterno:

[Ver todos](#)

Emérito	Edo. civil	Sit. Personal	Domicilio	Contactos	Antigüedad	Sit. Laboral	Padecimientos
NOMBRE DE EMÉRITO			No ingresado	No ingresado			No ingresado
NOMBRE DE EMÉRITO			No ingresado	No ingresado			No ingresado
NOMBRE DE EMÉRITO		No ingresado	No ingresado	No ingresado	No ingresado	No ingresado	No ingresado
NOMBRE DE EMÉRITO		No ingresado	No ingresado	No ingresado	No ingresado	No ingresado	No ingresado
NOMBRE DE EMÉRITO		No ingresado	No ingresado	No ingresado	No ingresado	No ingresado	No ingresado
NOMBRE DE EMÉRITO		No ingresado	No ingresado	No ingresado	No ingresado	No ingresado	No ingresado
NOMBRE DE EMÉRITO		No actualizado desde 26/06/2012	No ingresado				
NOMBRE DE EMÉRITO		No actualizado desde 28/06/2012	No ingresado				
NOMBRE DE EMÉRITO		No actualizado desde 28/06/2012	No ingresado				

1 2 3

Pólizas próximas a expirar
(Para mayor detalle, consultar la pantalla de pólizas contratadas)

Póliza	Tipo de póliza	Compañía	Fecha inicio	Fecha fin
poliza vida 1	De vida	001	6/08/2012	2/12/2012

Fig. 23 - Página “Alertas”

Página “Reportes”

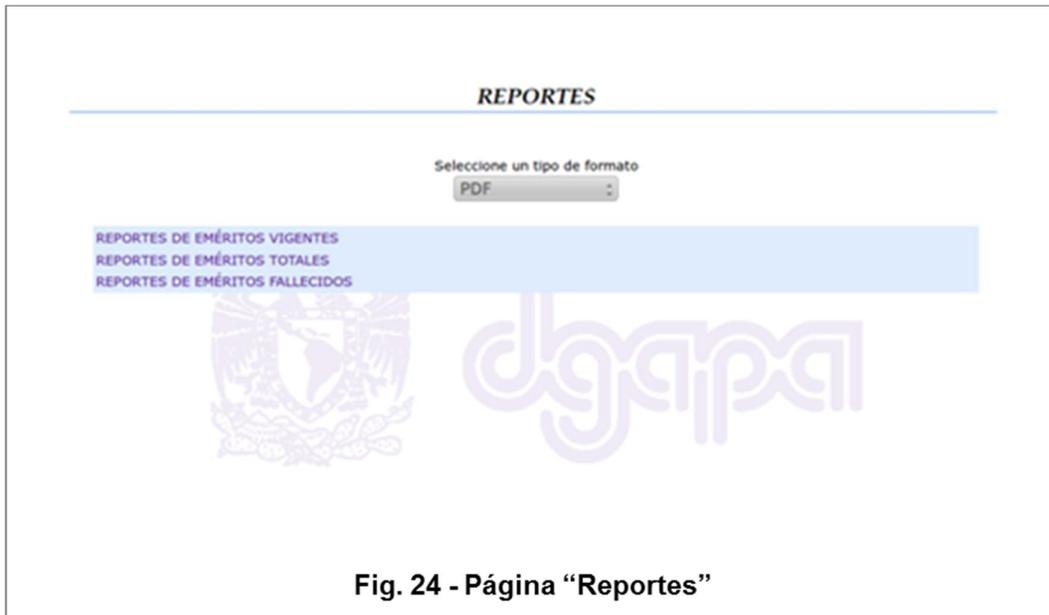


Fig. 24 - Página “Reportes”

Página “Gráficas”

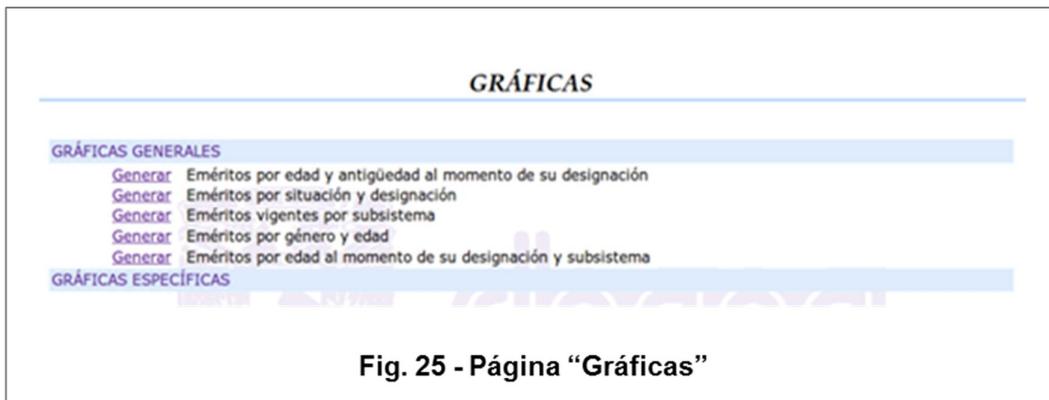


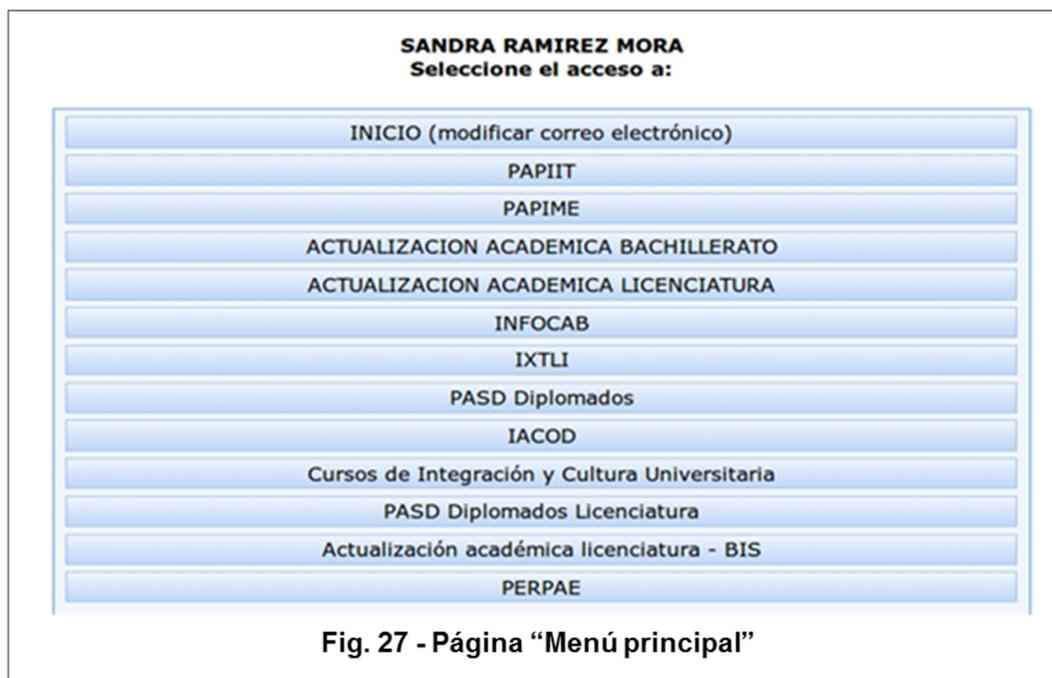
Fig. 25 - Página “Gráficas”

Página “Registro”



Fig. 26 - Página “Registro”

Página “Menú principal”



GLOSARIO

Alter: Sentencia utilizada para modificar las características de estructuras de bases de datos.

Combo: Elemento utilizado para desplegar datos en una aplicación de software.

Commit: Sentencia que finaliza una transacción.

Create: Sentencia utilizada para crear estructuras en una base de datos.

CSS (Cascading Style Sheets): Lenguaje utilizado para definir la presentación de páginas web.

DAO (Data Access Objects): Estructura utilizada para acceso a datos por medio de una interface.

DDL (Data Definition Language): Es el lenguaje utilizado para definir estructuras de bases de datos.

DCL (Data Control Language): Lenguaje utilizado para controlar el acceso a bases de datos.

Delete: Sentencia para eliminar datos de estructuras de bases de datos.

DML (Data Manipulation Language): Lenguaje que permite la consulta y manipulación de una base de datos.

Drop: Sentencia utilizada para eliminar un elemento almacenado en la base de datos.

DTO (Data Transfer Object): Estructuras utilizadas para transferir datos entre componentes de software.

Framework: Tecnología que da soporte a un software.

Gradle: Herramienta para organizar proyectos de software.

Grant: Sentencia que permite asignar privilegios de acceso a usuarios.

Grid: Elemento para desplegar datos que cuenta con una estructura bidimensional.

HTML (HyperText Markup Language): Lenguaje utilizado para la creación de páginas web.

IDE (Integrated Development Environment): Entorno de desarrollo para editar y generar código.

Insert: Sentencia para insertar datos en estructuras de bases de datos.

OMG (Object Management Group): Es un consorcio que se encarga de establecer estándares de la tecnología orientada a objetos.

Plug-in: Complemento de una aplicación para brindarle cierta funcionalidad.

POJO (eXtensible Markup Language): Clases simples en Java que no dependen de un framework.

POM (Project Object Model): Modelo para describir los componentes de un proyecto generado con Maven.

Query: Sentencia mediante la cual se puede acceder a una base de datos.

Revoke: Sentencia para suprimir privilegios de acceso a usuarios.

Rollback: Sentencia que aborta una transacción.

Select: Sentencia para realizar consultas sobre una base de datos.

Trigger: Procedimiento que se ejecuta en una base de datos cuando se cumple una acción.

Truncate: Sentencia para borrar el contenido de tablas.

Update: Sentencia para modificar datos de una tabla.

WAR (Web Application Archive): Archivo para organizar los elementos de una aplicación en Java.

WWW (World Wide Web): Sistema de distribución de información accesible a través de internet.

XML (eXtensible Markup Language): Lenguaje de marcas que permite definir y manipular información en diferentes formatos.

BIBLIOGRAFÍA Y REFERENCIAS **ELECTRÓNICAS**

PRESSMAN Roger S. Ingeniería del Software. Quinta edición. McGraw-Hill. 2002. ISBN: 84-481-3214-9.

SOMMERVILLE Ian. Ingeniería del Software. Primera edición. Pearson. 2005. ISBN: 8478290745.

BOOCH Grady, RUMBAUGH James, JACOBSON Ivar. The UML user guide. Primera edición. Addison Wesley. 1998. ISBN: 0-201-57168-4. 512 p.

LARMAN, Craig. UML y patrones. Primera edición. Prentice Hall. 507 p.

PEREZ Gustavo. Aprendiendo Java y programación orientada a objetos. 2008.

MUGHAL Khalid A., RASMUSSEN Rolf W. A programmer's guide to Java SCJP certification. Tercera edición. Pearson. 2008.

Sitio de la DGAPA

<http://dgapa.unam.mx/>

[Consulta: 23 de febrero de 2012]

Consulta del Estatuto del Personal Académico de la UNAM

<http://info4.juridicas.unam.mx/unijus/unv/3/default.htm?s=unjts>

[Consulta: 09 de marzo de 2012]

Sitio de la Dirección General de Administración Escolar de la UNAM para consulta del Reglamento del Reconocimiento al Mérito Universitario de la UNAM

https://www.dgae.unam.mx/normativ/legislacion/mod_merito_universitario2.html

[Consulta: 09 de marzo de 2012]

Sitio de UML

<http://www.uml.org/>

[Consulta 14 de abril de 2012]

Sitio de OMG para consulta de UML y sus versiones

<http://www.omg.org/spec/UML/>

[Consulta 10 de junio de 2012]

Repositorio de *PostgreSQL* versión 8.2.4

<http://www.postgresql.org/docs/8.2/static/datatype.html>

[Consulta 22 de Julio de 2012]