
Capítulo 1 Marco teórico y antecedentes del proyecto

1.1 Marco teórico.

Es importante indicar que la base fundamental para desarrollar un proyecto de sistemas de principio a fin es trabajar utilizando el apoyo de una metodología.

Primero se reúne la información acerca del objetivo del requerimiento para dar inicio a un análisis, para esto se deben hacer reuniones con los usuarios responsables del área que solicita el proyecto en donde se recopila la información de que procesos son los que llevan a cabo, las funciones que tiene cada persona que interviene en cada paso de los procesos y complementando esto con sugerencias del área de sistemas se define que es lo que debe hacer el sistema.

El diseño del sistema, en esta etapa se deben definir los archivos, las relaciones entre éstos y los programas que conforman el sistema para poder identificar qué recursos se necesitan en este caso el equipo de trabajo y el tiempo estimado para realizar el proyecto.

El desarrollo inicia con la creación de los archivos de bases de datos del sistema, éstos serán el repositorio de la información temporal y permanente relacionada debidamente entre sí para un correcto funcionamiento.

La creación de los programas será utilizando la técnicas de la programación estructurada creada por Bohm-Jacopini, quien plantea que todo programa puede escribirse utilizando únicamente las tres instrucciones de control básicas, secuencia, instrucción condicional e iteración (bucle de instrucciones) con condición al principio.

Una estructura de programa es secuencial si las instrucciones se ejecutan una tras otra, a modo de secuencia lineal, es decir que una instrucción no se ejecuta hasta que finaliza la anterior.

La estructura selectiva permite que la ejecución del programa se bifurque a una instrucción (o conjunto) u otra/s, según un criterio o condición lógica establecida, sólo uno de los caminos en la bifurcación será el tomado para ejecutarse.

Un bucle iterativo o iteración de una secuencia de instrucciones, hace que se repita su ejecución *mientras* se cumpla una condición, el número de iteraciones

normalmente está determinado por el cambio en la condición dentro del mismo bucle [1].

1.1.1 Programación estructurada

El case SNAP provee de las estructuras necesarias para crear programas sencillos de comprender, con lo que se disminuye así el tiempo requerido en las etapas de programación y mantenimiento.

El flujo de la lógica de un módulo siempre empieza con la primera instrucción, y sigue hacia abajo con las demás instrucciones una tras otra, excepto en caso de una instrucción condicional (SI...SINO...FIN o CASO...CASO...SINO...FINCASOS), con la que se puede escoger que se ejecute un bloque de instrucciones, otro bloque o ninguno, o en un ciclo (MIENTRAS...FIN o REPITA...FIN) donde se repite el bloque de instrucciones.

1.1.2 Programación modular

La filosofía de programación de SNAP es completamente modular; es decir, que los procesos deben descomponerse en sub-procesos más pequeños y sencillos de comprender, para facilitar la programación y el mantenimiento. De esta manera, se construirán programas pequeños que ejecuten otros programas para simplificar los procesos. Así mismo, un programa debe componerse de varios módulos o rutinas de proceso que ejecuten partes específicas.

Todo programa de SNAP empieza en el módulo 'MAIN', a partir del cual se puede llamar a los demás módulos del programa (instrucción EJECUTAR módulo).

Se recomienda crear un módulo de inicialización del programa, donde se realicen todas aquellas operaciones iniciales, necesarias para realizar el resto del programa, como por ejemplo, inicialización de variables, lectura de información de la base de datos, etc. Y luego, otro módulo donde se realice el cuerpo del programa, el cual puede utilizar otros módulos. Estos dos módulos deberán ser llamados por el módulo MAIN. Adicionalmente, el módulo MAIN se encarga de obtener la información especial que se le pasa al programa para su ejecución por medio de parámetros.

También se puede llamar a cualquier programa que esté en el repositorio central o a cualquier programa existente en el sistema, sin importar el lenguaje en que fue programado.

Por último, existen procesos de uso común que se han programado como funciones SNAP, y que están a disposición del usuario para que se utilicen en cualquier programa (instrucción función). Además, el usuario puede programar nuevas funciones e incorporarlas al repositorio central.

Para garantizar que el producto final cumplirá con las expectativas del usuario es necesario realizar pruebas de los programas en conjunto con éstos.

La liberación al ambiente de producción será en el momento que las pruebas sean satisfactorias.

El soporte a la producción consiste en detectar nuevas necesidades o formas nuevas y más eficientes para explotar el uso del sistema y la administración de su información. [2]

La plataforma sobre la que trabajé haciendo funciones de análisis y desarrollo de sistemas fue en un equipo IBM AS400 que posteriormente se ha convertido en el equipo Iseries 5 de IBM, los proyectos que se hicieron durante el tiempo que trabajé con éste en la universidad durante 16 años fueron elaborados utilizando una herramienta case que se llama SNAP junto con las herramientas propias del equipo para la explotación de la información como son el SQL400, Query400, DFU (Data File Utility) que sirven para la consulta y el mantenimiento de información de los archivos directamente sin necesidad de programas.

El SNAP proporciona un ambiente de trabajo integrado para el desarrollo y mantenimiento de sistemas en el AS/400 de IBM.

Esta herramienta está orientada a simplificar el proceso de construcción de programas, acelerando substancialmente la productividad de desarrollo sin perder la flexibilidad y control sobre los proyectos. El esfuerzo de mantenimiento de los programas generados se minimiza considerablemente, debido al apoyo en la documentación técnica y estructura del repositorio y al poder que aportan la herramientas de regeneración y reconstrucción.

El SNAP proporciona especificaciones de muy alto nivel para definir una estructura de datos por programa, diseñar pantallas, especificar reportes y los módulos lógicos requeridos mediante la utilización de este novedoso lenguaje de cuarta generación,

que cuenta con un conjunto completo de macro y micro instrucciones estructuradas en lenguaje español.

En la figura 1.1 se muestra la metodología que sirve para optimizar el desarrollo de cualquier programa y obtener un significativo incremento de la productividad de un programador.

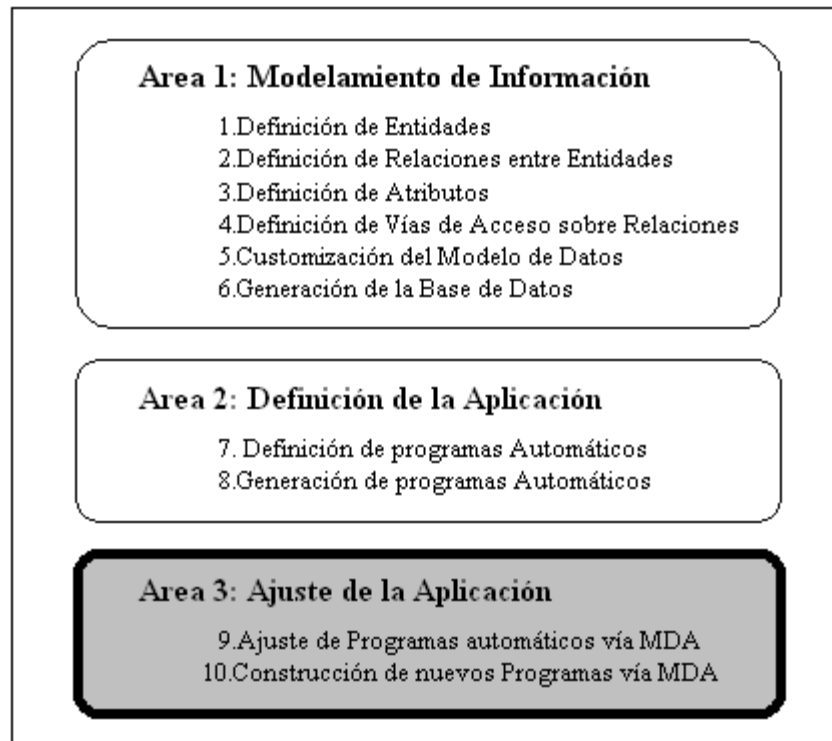


Figura 1.1 Metodología de desarrollo snap.

La herramienta case SNAP posee poderosas instrucciones para acceder y mantener la integridad de la base de datos. También existe independencia en la definición de los archivos y los programas, ya que éstos utilizan las entidades definidas en el modelo de datos. De esta manera, se aprovechan las características propias del equipo en el manejo y administración de la base de datos.

El modelaje de la información nace como una necesidad derivada del proceso de diseño de la base de datos. [3]

1.1.3 Definición de la Base de Datos

El proceso de diseño de una base de datos se compone de tres niveles de definición y de un nivel físico donde se apoyan los restantes.

El nivel físico es donde se establecen todas las características físicas de los datos que van a ser almacenados en los dispositivos físicos de almacenamiento. Se trabaja a nivel de bytes, direcciones físicas, capacidades, algoritmos de acceso generalmente implantados en hardware o firmware (circuitería o microcódigo). En la mayoría de los sistemas, este nivel es tarea del sistema operativo

El nivel interno es donde se define la estructura de almacenamiento de los datos y la estrategia de acceso a éstos. Se trabaja a nivel de campo, registro, archivo, índice, estructura de datos, direcciones lógicas o algoritmos de acceso generalmente implantados en software. Esta definición se conoce como esquema interno. En el AS/400 esta definición se realiza por medio del archivo fuente de descripción de datos (DDS), que maneja una estructura de datos relacional

El nivel conceptual es donde se definen las características y relaciones lógicas entre los datos que se desean almacenar. Se trabaja a nivel de atributo, entidad, relación, vía de acceso. Esta definición se conoce como esquema conceptual.

El nivel externo es donde se definen las características de los datos almacenados tal y como las ve un usuario exterior; por eso se le conoce como vista lógica o vista externa. Se trabaja a nivel de información. En muchos sistemas administradores de base de datos no se hace diferencia entre los dos últimos niveles, identificando el nivel externo con el conceptual. En este caso cada vista es una forma diferente de manejar la misma información. Por otro lado, si se tiene que definir por partes este nivel, entonces cada vista será una parte determinada o subconjunto de la totalidad del esquema conceptual y cada definición se conocerá como sub-esquema.

Generalmente, en un sistema administrador de base de datos relacional como el del AS/400, una entidad se convierte físicamente en un archivo y una ocurrencia representa un registro dentro de ese archivo.

Las estructuras de datos determinan de dónde se obtiene la información y dónde se almacenará, para que el programa la pueda procesar.

Si se necesita acceder a la base de datos para obtener o almacenar información, se deben seleccionar las entidades involucradas en cada programa; si se desea almacenar datos temporalmente para el proceso, se definen variables y matrices de trabajo.

El Modelo de Datos representa el primer paso dentro de la metodología de desarrollo del SNAP. Dentro de este ambiente de trabajo se agrupan todas las herramientas necesarias para definir, documentar y generar la representación conceptual de la empresa u organización, y obtener como resultado, la base física de los datos.

Se utiliza el modelo Entidad-Relación, el cual fue desarrollado originalmente por Peter Chen a partir de 1976 y que se ha ido perfeccionando con los años.

Este modelo representa toda una filosofía de desarrollo y se basa en la percepción de un mundo compuesto por Objetos (Entidades) con características propias (Atributos) y de asociaciones entre esos objetos (Relaciones).

La definición de las entidades: En SNAP se identifican como entidades a los archivos de la base de datos que integran el sistema, una entidad es cualquier objeto del mundo real que tiene vida propia o que puede distinguirse de otros objetos del mismo tipo y del cual se desea guardar información. Puede ser una representación de algo concreto como un auto o una persona, o de algo abstracto como un concepto o una fecha.

Los atributos son las propiedades que caracterizan cada entidad y representan la información que se desea almacenar con respecto a la entidad a la cual pertenecen por lo tanto cada atributo se va a convertir en un campo del archivo de la entidad, el dominio de un atributo determina las características de los valores válidos que puede almacenar. Establece el tipo o formato físico del dato, es decir, si permite valores numéricos o alfanuméricos, también establece un rango de valores o un conjunto de valores permitidos y puede establecer una verificación especial.

Definición de relaciones entre entidades: Una relación establece las reglas de asociación entre las ocurrencias de dos entidades. Como ocurre en el mundo real, los objetos (entidades) cuentan con toda una serie de asociaciones (relaciones) entre ellos; por ejemplo, un artículo está asociado a la bodega en la cual está almacenado, una factura está relacionada con el cliente respectivo, etc. Estas relaciones son las que permiten la existencia de un sistema real, es decir, que todas las partes interactúen y afecten al todo. Por lo anterior, nunca debería existir una entidad o

grupo de entidades aislado dentro de un modelo de datos, ya que al no interactuar con otras partes no pertenecen directamente al sistema total.

La cardinalidad de la relación determina el tipo de asociación entre las ocurrencias de dos entidades. Esta permite que existan relaciones de uno-a-uno, de uno-a-muchos (o muchos-a-uno) y de muchos-a-muchos.

Relación Uno-a-Uno. Una relación de uno-a-uno se define cuando una ocurrencia de una entidad se relaciona tan sólo con una ocurrencia de la otra entidad y viceversa.

Relación Uno-a-Muchos. Una relación de uno-a-muchos se define cuando una ocurrencia de una entidad se relaciona con muchas ocurrencias de la otra entidad, pero una ocurrencia de ésta se relaciona tan sólo con una ocurrencia de la primera.

Relación Muchos-a-Muchos. Una relación de muchos-a-muchos se define cuando una ocurrencia de una entidad se relaciona con muchas ocurrencias de la otra entidad y viceversa.

Redundancia. La redundancia es una repetición de la misma información en dos o más lugares diferentes. Esta repetición puede causar errores de inconsistencia si se actualiza esa información en un lugar y no en otro. Por esta razón, es necesario que toda redundancia sea controlada, es decir, que permanezca consistente en todo momento.

Normalización. Una entidad está normalizada si, y sólo si, no posee atributos repetitivos, y cada campo no-clave es mutuamente independiente pero depende directamente de una sola clave primaria.

Definición de atributos

Campos clave. Una clave es un campo o una combinación de campos cuya información tiene una función especial dentro del archivo al que pertenece.

Clave Primaria. Una clave primaria es un campo o una combinación de campos que identifican en forma única cada ocurrencia de la entidad y por tanto, no habrá más de una ocurrencia con los mismos valores, es decir, no tendrá duplicados ni tampoco podrá tener un valor nulo.

Sin embargo, es posible que dentro de una entidad haya diferentes campos o conjuntos de campos que cumplan con la definición anterior; en este caso, esos campos se conocen como claves candidatas (candidatas a convertirse en clave

primaria). Sin embargo, sólo uno de ellos se convertirá en clave primaria y el elegido será el más conveniente por su menor tamaño o porque nos asegura que no existirán dos claves repetidas. [4]

1.2 Antecedentes.

El maestro en mercadotecnia coordinador del departamento del centro de idiomas en primera instancia solicitó Jefe de aplicaciones de gestión escolar del área de desarrollo de sistemas perteneciente a la coordinación de sistemas, el apoyo para crear una aplicación que agilizará los trámites y un funcionamiento más efectivo por parte de su personal para poder atender a los alumnos en su inscripción, controlar el pago y adeudos parciales, asignar los grupos, realizar la captura de calificaciones finales y crear una base de datos histórica para integrar al final de los cursos los idiomas con sus respectivas calificaciones.

A partir de esto se realizan juntas para definir los requerimientos del área solicitante, se definen los archivos y programas que conformaran el sistema y se establece la duración de la primera etapa del proyecto en 5 meses con revisiones periódicas de los avances, el proceso de inscripción y control de la información de los alumnos del centro de idiomas se realizaba de forma manual y mediante hojas de cálculo (EXCEL) generando inconformidad por parte de la población estudiantil debido a la pérdida de tiempo que esto implicaba.

El alumno tenía que acudir personalmente al centro de idiomas a ver las listas que se pegaban en las paredes del edificio, estas contenían los horarios de los distintos niveles de idiomas, lo cual permitía tener acceso visualmente a la oferta de Idiomas para seleccionar los de su interés, el alumno debía llenar una solicitud que contenía el tipo de alumno (Interno, externo o empleado), su clave de alumno en el caso de los internos, clave de empleado en caso de empleados y los externos no tenían clave, nombre, carrera, idioma y horario solicitado.

Era necesario formarse en una fila en el mostrador para esperar a ser atendido por una de las dos personas asignadas para esta labor por la coordinación del centro de idiomas, sin embargo al aglomerarse la gente y aumentar excesivamente las filas, se generaba molestia e inconformidad por parte de los alumnos y a su vez sensación de malestar y presión para el personal que los atendía por que el servicio se volvía lento e ineficiente.

Surgía en ese momento la necesidad de que otras personas acudieran a brindar apoyo para atender a los alumnos separándose de sus funciones incluso el mismo coordinador para poder incrementar el ritmo de atención

Se ocasionaban problemas de otra índole para los empleados como es tener que quedarse a trabajar después de su horario para poder organizar lo que se acumuló durante el día y además ponerse al corriente en sus labores habituales.

Al siguiente día se volvía a iniciar nuevamente el mismo ciclo incluso incrementándose día a día la necesidad de atención de los alumnos ya que por lo general los últimos días es cuando más alumnos acuden por diversas razones como que estaban en espera de obtener el dinero para inscribirse o también que se encontraban simultáneamente en el proceso de inscripción de las materias regulares correspondientes al semestre de su licenciatura.

Esto también ocasionaba que se tuvieran que recorrer las fechas de aceptación de solicitudes para poder captar a todos los interesados en cursar algún nivel de un idioma.

Después las personas del mostrador tenían que realizar una captura manual de las solicitudes en Excel y entregar varios archivos, uno por cada idioma, al responsable administrativo quien tenía que unificar la información y para poder organizar los grupos tenía que repartir a los alumnos en los grupos existentes, a veces era necesario cambiar algunos alumnos de horario y se quedaba pendiente confirmar con éstos si podían cursar en el nuevo grupo asignado.

A partir de ahí se revisaba y se imprimían las listas definitivas de los grupos, que se pegaban en las paredes del centro de idiomas.

Posteriormente el proceso de cambios se hacía cuando a algún alumno no le era posible cursar en el grupo nivel u horario asignado, se tenía que estar tachando y/o agregando en su caso sobre estas listas y modificando posteriormente el documento electrónico de Excel, lo cual traía como consecuencia un problema muy fuerte de inconsistencia y falta de integridad de la información cuando existía alguna confusión u olvido de parte de la persona que llevó a cabo el trámite.

Las calificaciones finales eran capturadas manualmente por cada profesor sobre las listas impresas, así como las correcciones y éstas se entregaban directamente al responsable administrativo, que tenía que actualizar sobre el documento de Excel, volviéndose necesario realizar revisiones continuas y exhaustivas de la información, para tener un registro histórico de las calificaciones al final de los cursos todo el

personal se dedicaba a realizar una captura manual en Excel que se integraba también por el responsable administrativo.

La explotación de esta información se volvía complicada ya que es necesario tener estos datos en el momento que los alumnos desean inscribirse al siguiente nivel para poder validar que no vuelvan a cursar el mismo nivel nuevamente en caso de haberlo aprobado o que no cursen un nivel sin haber aprobado el inmediato anterior del mismo idioma.

Lo deseable en este caso es que los alumnos puedan personalmente hacer el trámite de seleccionar el horario de un nivel de algún idioma y acudir a caja a pagar el monto requerido de acuerdo al tipo de alumno, siendo el sistema el que realice las validaciones necesarias, de seriación, cupo del grupo y el cálculo del costo.