



**FACULTAD DE CIENCIAS
DE LA COMPUTACIÓN**

***"Sistema Web de captura, evaluación y registro, de la
estructura de representación ante órganos electorales
de un partido político"***

TESIS

PARA OBTENER EL TÍTULO DE:

Licenciada en Ciencias de la Computación

PRESENTA:

Deyanira Pérez González

ASESORA:

DRA. MARÍA JOSEFA SOMODEVILLA GARCÍA

NOVIEMBRE DE 2014



RESUMEN

Este proyecto tiene como fin facilitar la manipulación de la información de los representantes de un partido político ante los órganos electorales, para poder tener control de los datos requeridos del representante y así realizar varias tareas que van desde la supervisión del avance de la estructura de representantes hasta el registro de estos ante el órgano oficial.

Aprovechamos la llegada de las metodologías ágiles a la ingeniería de software; para aplicarlas en el desarrollo de este proyecto, una de ellas llamada SCRUM la cual trabaja con iteraciones de desarrollo llamadas "sprints". En esta metodología al finalizar cada sprint se debe entregar un producto funcional, por lo que el número de iteraciones necesarias estuvieron dadas por el cumplimiento en la especificación de requerimientos.

En todo desarrollo de software es necesario comprender el problema (Análisis), plantear una posible solución (diseño), llevar a cabo la solución planteada (implementación) y comprobar que el resultado obtenido es correcto (pruebas). Por tanto, combinamos las etapas básicas del ciclo de vida del desarrollo de software, con esta metodología.

En la etapa del Análisis entendimos qué es exactamente lo que tiene que hacer el sistema, esto lo plasmamos a través de la especificación de requerimientos y algunos diagramas generados en UML. En la fase del Diseño representamos las características del sistema que permitieron implementarlo de forma efectiva. Posteriormente en la fase de Implementación seleccionamos las herramientas que consideramos adecuadas, por su entorno de desarrollo Dreamweaver nos facilitó el trabajo, además los lenguajes de programación PHP y HTML que consideramos apropiados para el tipo de sistema que construimos. Por último en la etapa de pruebas tuvimos por objetivo detectar los errores que cometimos en las 3 etapas anteriores.

Como resultado de este proyecto se tiene un producto funcional que cumple con la especificación de requerimientos surgidos en la etapa de Análisis.

Contenido

RESUMEN	2
CAPÍTULO1 INTRODUCCIÓN.....	5
1.1 ANTECEDENTES.....	7
CAPÍTULO 2 MARCO TEÓRICO.....	8
2.1 METODOLOGÍAS ÁGILES	8
2.2 SCRUM	8
2.2.1 Roles.....	8
2.2.2 Reuniones	8
2.2.3 Componentes.....	9
2.3 MODELO CLIENTE - SERVIDOR.....	10
2.3.1 Cliente.....	11
2.3.2 Servidor.....	11
2.3.3 Beneficios.....	12
2.3.4 Características.....	12
2.3.5 Clasificación	12
2.4 BASES DE DATOS.....	14
2.4.1 Modelo relacional de datos	14
2.4.2 Relaciones entre tablas.....	17
2.4.3 Modelado Entidad-Relación (ER)	17
2.4.4 Modelado ER-Extendido (ERE).....	19
2.4.5 Lenguaje estructurado de datos (SQL).....	21
CAPÍTULO 3 ANÁLISIS DEL SISTEMA.....	24
3.1 OBJETIVOS DEL PROYECTO	24
3.2 PRODUCT BACKLOG	24
3.1 1 ^{er} SPRINT	26
3.3.1 Backlog del 1 ^{er} Sprint	26
3.3.1 Diagrama de contexto.....	27
3.3.2 Diagrama de casos de uso	29
3.3.3 Escenarios de casos de uso.....	30
3.3.4 Diagrama de clases	37

4.3.5 Diagrama de secuencia	38
CAPÍTULO 4 DISEÑO DEL SISTEMA	39
4.1 DISEÑO ARQUITECTÓNICO DEL SISTEMA	39
4.2 DISEÑO DE LA BASE DE DATOS	41
4.2.1 Diseño Conceptual	41
4.2.2 Diseño Lógico	43
4.2.3 Diseño Físico	48
4.3 DISEÑO DE LA APLICACIÓN	48
4.3.1 Diseño de la Interfaz de Usuario	49
CAPÍTULO 5 IMPLEMENTACIÓN DEL SISTEMA	51
5.1 IMPLEMENTACIÓN DE LA BASE DE DATOS	51
5.2 IMPLEMENTACIÓN DE PROGRAMAS DE APLICACIÓN	53
5.3 SEGURIDAD	55
5.3.1 Protección al Sistema	55
5.3.2 Protección de la BD del Sistema	58
5.4 2 ^{do} Sprint	64
5.5 3 ^{er} Sprint	64
CAPÍTULO 6 PRUEBAS	65
6.1 CONVERSIÓN Y CARGA DE DATOS	65
6.2 PRUEBAS	67
6.2.1 Captura de representante	67
6.2.3 Eliminación de representante	71
6.2.4 Consulta de representante	72
6.2.5 Estadísticas avance de estructura por distrito	74
6.2.5 Estadísticas avance de estructura Estatal	76
6.2.6 Generación de Oficio de Acreditación	78
6.3 Conclusión de las pruebas.	78
CONCLUSIONES	79
TRABAJO FUTURO	81
BIBLIOGRAFÍA	82
APÉNDICE DEL MARCO TEÓRICO	83

CAPÍTULO1 INTRODUCCIÓN

Para tener un mejor entendimiento del propósito del presente proyecto, comenzaremos por explicar a grandes rasgos acerca de cómo se estructura el Instituto Federal Electoral (IFE) y el Instituto Electoral Estatal del Estado de Puebla (IEE), en tiempos de proceso electoral.

El IFE es un organismo público, autónomo y permanente que organiza las elecciones federales, es decir las elecciones de Presidente de la República, Diputados y Senadores que integran el Congreso de la Unión. Para su correcto desempeño este comienza la preparación de las elecciones con antelación desde el año anterior al del proceso electoral. El Proceso electoral inicia con la Instalación del Consejo Local en noviembre, a mediados de enero se instalan los Consejos Distritales y por último, el primer domingo de julio se instalan las Mesas Directivas de Casilla para la recepción del voto ciudadano. Esta estructura tiene una forma piramidal, el Consejo Local, que sería la cúspide de la pirámide. Con respecto a los Consejos Distritales en Puebla se instalan 16 y 7 mil Mesas Directivas de Casilla aproximadamente para todo el estado [1]. El IEE tiene una forma de trabajo y una estructura muy parecida a la del IFE, los niveles son: Consejo General (1); Consejos Distritales (26) y Mesas Directivas de Casilla (7 mil aproximadamente) [2].

Tanto el Código de Federal de Instituciones y Procedimientos Electorales (COFIPE), como el Código de Instituciones y Procesos Electorales del Estado de Puebla (COIPEP), permiten a los Partidos Políticos (PP) tener representantes en estos niveles de estructura del IFE y del IEE respectivamente. Estos representantes deben registrarse oficialmente ante estas instituciones, en una fecha límite que estipulan estos códigos. Por lo que un PP también tendrá una estructura de forma piramidal para poder supervisar los trabajos de la jornada electoral.

En el presente proyecto se desarrolló un Sistema Web de tipo cliente-servidor que facilitará la recolección de información de los representantes electorales de un PP, para llevar un seguimiento y poder hacer una evaluación del avance en la construcción de esta estructura del partido político, así como, de la prevención de representantes repetidos, en el estado.

La metodología de desarrollo de proyectos que se usó en el presente trabajo se denomina "Scrum", que es una técnica clasificada dentro de las metodologías ágiles de desarrollo de software. Por ser Scrum una metodología ágil orientada al desarrollo de proyectos no cuenta con técnicas para la elaboración de diagramas que auxilien en el análisis del sistema, por lo que nos auxiliaremos de la metodología UML para desarrollar los diagramas de casos de uso, de clase, de objeto, de actividad y de secuencia. Para la elaboración de estos diagramas se usó el software Visual Paradigm, que es una herramienta basada en UML.

Para la programación del sistema se utilizaron los lenguajes PHP y JavaScript, para la creación y manipulación de la Base de Datos se utilizará el Gestor de Bases de Datos MySQL y por último para el hospedaje del sistema se utilizó el servidor de Páginas Web Apache.

El presente trabajo está distribuido de la siguiente forma:

Capítulo I.- Introducción al proyecto y antecedentes que originan la necesidad de una solución de software, para el control de representantes de un partido político.

Capítulo II.- Trata del marco teórico del desarrollo del sistema, entre los fundamentos teóricos encontraremos: las metodologías ágiles, Bases de Datos, la Arquitectura Cliente Servidor y PHP.

Capítulo III.- Da a conocer el objetivo general y los particulares del desarrollo del sistema, de los requerimientos del mismo y de la interpretación de estos a través del diagrama de casos de uso y escenarios.

Capítulo IV.- Este lo dedicamos en la primera parte a la elaboración del diseño del sistema, a través de los diagramas de clase y de secuencias. En la siguiente parte se habla del diseño de la Base de Datos que se va a utilizar para el sistema, misma que pasará por el modelado conceptual, el de Entidad - Relación, el Relacional y por la normalización de la misma. Para la última parte hablaremos del diseño de la aplicación en la que especificaremos las interfaces de usuario y los programas de aplicación.

Capítulo V.- En este capítulo hablaremos de la implementación de la base de datos, de los programas de aplicación y de la seguridad en el sistema y de la base de datos en uso.

Capítulo VI.- Dedicamos este capítulo a las pruebas del sistema, en la primera parte del capítulo mencionamos la importancia de cargar datos en la Base de Datos para poder realizar las pruebas.

1.1 ANTECEDENTES

Existen dos tipos de procesos electivos de candidatos a puestos populares o de candidatos de elección popular, estos son el Local y el Federal. El Proceso Federal se lleva a cabo cada 6 años para la elección de Presidente de la República, Senadores y Diputados Federales. Cada 3 años se llevan a cabo elecciones intermedias para Diputados Federales [1]. Para el Proceso Local, en donde se eligen Gobernador, Diputados Locales y Miembros de los Ayuntamientos, el tiempo es también cada 6 años y para las elecciones intermedias, es decir donde se eligen Diputados Locales y Miembros de los Ayuntamientos, cada 3 años [2].

Las elecciones más próximas serán las de Diputados Federales, que se llevarán el primer domingo de julio de 2015. Dentro de los preparativos de un Partido Político, para elecciones Locales o Federales, se encuentra el "armar" dos tipos de estructuras: La de promoción al voto y la de representantes ante los órganos electorales. La de promoción se encarga de la propaganda y promoción del voto en favor del partido político. La de representantes ante los órganos electorales se divide en 3 categorías: Representante ante Consejo Distrital (RCD), Representante General (RG) y Representante ante Mesa Directiva de casilla (RMDC) y tienen distintas funciones. El número aproximado de Representantes varía de 15 mil en un proceso Local a 20 mil en un proceso Federal [5].

Por las razones anteriormente expuestas el manejo de información de la estructura electoral sin un sistema es complicado. Entre los datos que se capturan son nombre, clave de elector, edad, domicilio completo, cargo, etc.

Con la Reforma a la Constitución del Estado de Puebla y al Código de Procedimientos Electorales del mismo, se empata el proceso **Local** y el **Federal**, para llevarse a cabo en el año 2018 [3] [4]. Por lo que estaríamos hablando de aproximadamente 35 mil representantes para este año.

CAPÍTULO 2 MARCO TEÓRICO

2.1 METODOLOGÍAS ÁGILES

Las "metodologías ágiles" son sistemas de desarrollo de software basado en valores, principios y prácticas básicas. Los productos que se diseñan mediante métodos ágiles se pueden desarrollar con rapidez. Dos palabras que caracterizan a un proyecto realizado mediante una metodología ágil son iterativo e incremental. Las etapas en el proceso de desarrollo ágil son exploración, planeación, iteraciones para la liberación de la primera versión, puesta en producción y mantenimiento. Algunos ejemplos de estas metodologías podrían ser: Programación Extrema (XP), Crystal, Feature Driven Development (FDD), Adaptive Software Development (ASD), Lean Development (LD) y SCRUM.

Dado que para el desarrollo de este proyecto, se eligió la metodología SCRUM, hablaremos de sus componentes y características:

2.2 SCRUM

Es una metodología ágil, que trabaja con iteraciones de desarrollo llamadas "Sprints" y recomienda se realicen cada 30 días. Los elementos que conforman el desarrollo del Scrum (figura 2.1) son [3]:

2.2.1 Roles

Propietario del producto: Quien determina las prioridades y define requisitos del producto.

Scrum Manager: Quien gestiona y facilita la ejecución del proceso.

Equipo: Quien o quienes construyen el producto.

Interesados: Quienes solo asesoran y observan.

2.2.2 Reuniones

Planificación del Sprint: Es la jornada de trabajo que se realiza antes del inicio de cada sprint y que sirve para determinar cuáles van a ser los objetivos que se deben cumplir en esa iteración.

Reunión diaria: Se realizan revisiones diarias de los avances del equipo en la que se hacen preguntas como ¿Qué hiciste ayer o en qué has avanzado?, ¿Qué vas a hacer hoy? y ¿Qué impedimentos tienes o podrías tener para cumplir con los compromisos en este sprint o en el proyecto?

Revisión del Sprint: Análisis y revisión del incremento generado.

2.2.3 Componentes

Product Backlog: Relación de requisitos del producto, que estará en evolución constantemente y el responsable de esta es el propietario del producto.

Sprint Backlog: Lista de compromisos del equipo para el Sprint suficientemente detallado para su fácil ejecución.

Incremento: Parte del producto desarrollado en un sprint, en condiciones funcionales. [4]

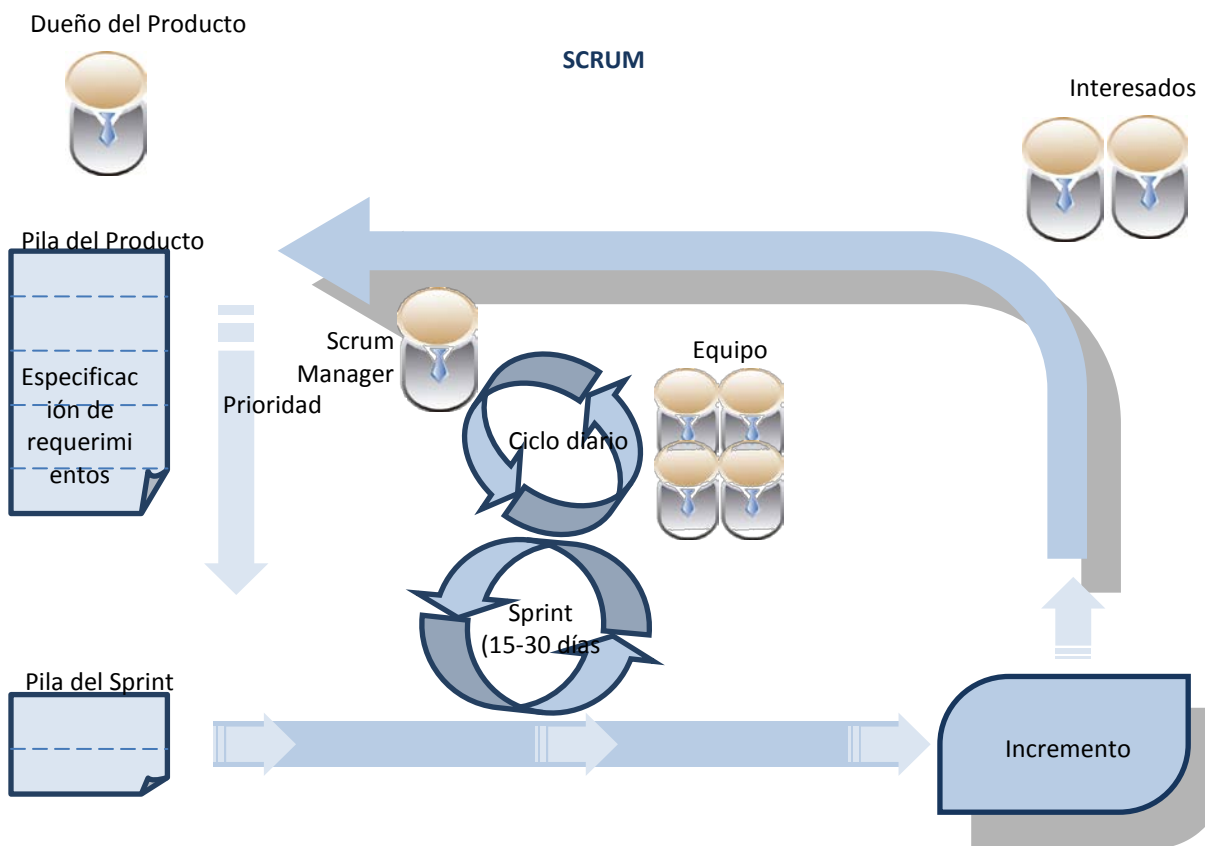


Fig. 2.1 Estructura Central del Scrum

2.3 MODELO CLIENTE - SERVIDOR

Es un modelo (figura 2.2) para el desarrollo de sistemas de información, en el que las transacciones se dividen en procesos independientes que cooperan entre sí para intercambiar información, servicios o recursos. Es el modelo más común entre aplicaciones en una red en donde interactúan cliente y servidor [5].

Modelo Cliente-Servidor

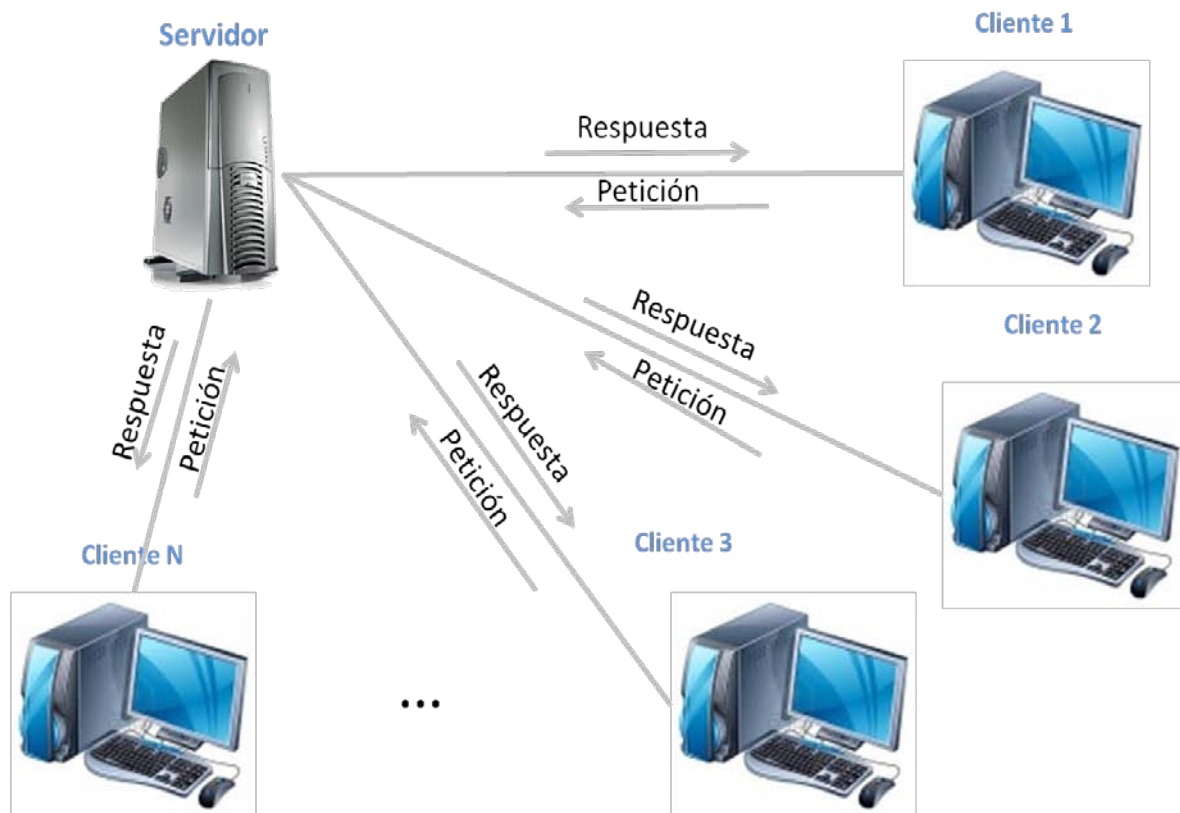


Fig. 2.2 Modelo Cliente Servidor

2.3.1 Cliente

Es el proceso que inicia el diálogo o solicita los recursos. También definido como un conjunto de Software y Hardware que invoca los servicios de uno o varios servidores. Regularmente éstos, los clientes, interactúan con el usuario en forma gráfica. Se pueden comunicar con procesos auxiliares que se encargan de establecer conexión con el servidor, enviar el pedido, recibir la respuesta, manejar las fallas y realizar actividades de sincronización y de seguridad.

Las características más importantes del cliente son que oculta al usuario el Servidor y la Red. Puede detectar e interceptar peticiones de otras aplicaciones y puede redireccionarlas. Se dedica a la cesión del usuario desde que inicia hasta que termina. El método que regularmente utiliza para llamar a los servicios es a través de Remote Procedure Calls (RPC).

La funciones más comunes que el cliente tiene son mantener y procesar todo el diálogo con el usuario. Es capaz de manejar las pantallas, menús de interpretación de comandos, entra de datos y validación. Además de poder procesar ayudas y recuperación de errores, también puede generar consultas e informes sobre alguna base de datos.

2.3.2 Servidor

Proceso que responde a las solicitudes. Definido también como un conjunto de Hardware y Software que responde a los requerimientos uno o varios clientes. Los Servidores proporcionan un servicio al cliente y devuelven los resultados. El servidor debe manejar los interbloques, la recuperación ante fallas, y otros aspectos afines. Por las razones anteriores, la plataforma asociada con los servidores es más poderosa que la de los clientes y deben de utilizarse PCs más poderosas, estaciones de trabajo o sistemas grandes. Además deben manejar servicios como administración de la red, mensajes, control y administración de la entrada al sistema ("login"), auditoría y recuperación y contabilidad.

Algunas funciones de los servidores son actualizar datos almacenados, administrar recursos compartidos, ejecutar transacciones, controlar el acceso concurrente a las bases de datos compartidas, enlazar las comunicaciones de otras redes de área local

o extensa. Según sus funcionalidades los servidores se pueden clasificar en: Servidor de Archivos, Servidor de Bases de Datos, Servidor de Comunicaciones, Servidor de Impresión, Servidor de Terminal y Servidor de Aplicaciones.

2.3.3 Beneficios

Mejoran el aprovechamiento de los sistemas de cómputo pues se reparte el trabajo entre varios servidores. Es posible reducir el tráfico en la red, pues en ocasiones no es necesaria una conexión dedicada todo el tiempo, pueden viajar los requerimientos en el momento que se necesite. Este tipo de modelo no requiere obligatoriamente trabajar bajo la misma plataforma, puede ser que el servidor esté bajo un sistema operativo diferente de los clientes, por lo que se permite el uso de interfaces gráficas variadas y versátiles. Acceso, almacenamiento y organización de datos.

Para que los clientes y los servidores puedan comunicarse se requiere una infraestructura de comunicaciones, que proporcione mecanismos básicos de direccionamiento y transporte. La mayoría de los sistemas Cliente/Servidor se basan en redes locales y por lo tanto utilizan protocolos no orientados a conexión, lo cual implica que las aplicaciones deben hacer las verificaciones.

2.3.4 Características

Entre las principales características de la arquitectura cliente/servidor, se pueden destacar las siguientes: El servidor presenta a todos sus clientes una interfaz única y bien definida, por lo que el cliente no necesita conocer la lógica del servidor, sólo su interfaz externa. El cliente no depende de la ubicación física del servidor, ni del tipo de equipo físico en el que se encuentra, ni de su sistema operativo. En caso de requerir cambios en el servidor esto implica pocos o ningún cambio en el cliente.

2.3.5 Clasificación

Según las funciones que tienen el cliente y el servidor, este modelo se puede clasificar en 5 niveles.

1^{er} Nivel: El cliente asume parte de las funciones de presentación de la aplicación, ya que siguen existiendo programas en el servidor, dedicados a esta tarea. Dicha distribución se realiza mediante el uso de productos para el "maquillaje" de las pantallas del mainframe. Esta técnica no exige el cambio en las aplicaciones orientadas a terminales, pero dificulta su mantenimiento. Además, el servidor ejecuta todos los procesos y almacena la totalidad de los datos. En este caso se dice que hay una presentación distribuida o embellecimiento.

2^o Nivel: La aplicación está soportada directamente por el servidor, excepto la presentación que es totalmente remota y reside en el cliente. Los terminales del cliente soportan la captura de datos, incluyendo una validación parcial de los mismos y una presentación de las consultas. En este caso se dice que hay una presentación remota.

3^{er} Nivel: La lógica de los procesos se divide entre los distintos componentes del cliente y del servidor. El diseñador de la aplicación debe definir los servicios y las interfaces del sistema de información, de forma que los papeles de cliente y servidor sean intercambiables, excepto en el control de los datos, que es responsabilidad exclusiva del servidor. En este tipo de situaciones se dice que hay un proceso distribuido o cooperativo.

4^o Nivel: El cliente realiza tanto las funciones de presentación como los procesos. Por su parte, el servidor almacena y gestiona los datos que permanecen en una base de datos centralizada. En esta situación se dice que hay una gestión de datos remota.

5^o Nivel: El reparto de tareas es como en el anterior y además el gestor de base de datos divide sus componentes entre el cliente y el servidor. Las interfaces entre ambos, están dentro de las funciones del gestor de datos y, por lo tanto, no tienen impacto en el desarrollo de las aplicaciones. En este nivel se da lo que se conoce como bases de datos distribuidas.

2.4 BASES DE DATOS

Desde el principio de los tiempos el hombre ha necesitado resguardar colecciones de datos con algún fin. Anteriormente la información se coleccionaba en grandes archivos de papel, después con la aparición de la computadora, se crearon programas que se encargaran de la administración de estos datos. Normalmente se archivaban en un formato diferente para cada aplicación, es decir, cada programador trabajaba con un formato de datos propio. Este tipo de sistemas "resolvían el problema temporalmente", la dificultad se encontró cuando aplicaciones diferentes querían compartir la misma información. Lo que indudablemente creaba redundancia e inconsistencia en la información.

Tratando de dar solución a estos inconvenientes, se creó el modelo jerárquico de datos, y otras estructuras de datos que no tuvieron mucho éxito. Fue en 1970 que Edward Codd, matemático de IBM, introdujo el término de "Bases de Datos" y propuso el modelo Relacional.

Con las bases de datos, se buscó independizar los datos de las aplicaciones, es decir, mantenerlos en espacios diferentes. Los datos residen en memoria y los programas mediante un sistema gestor de bases de datos (SGBD), manipulan la información. El SGBD recibe la petición por parte del programa para manipular los datos y es el encargado de recuperar la información de la base de datos y devolvérsela al programa que la solicitó. Cada programa requerirá de una cierta información de la base de datos, y podrá haber otros que utilicen los mismos datos, pero realmente residirán en el mismo espacio de almacenamiento y los programas no duplicarán esos datos, si no que trabajarán directamente sobre ellos concurrentemente. Aunque la estructura de la base de datos cambiara, si los datos modificados no afectan a un programa específico, éste no tendrá por qué ser alterado.

Es importante tener claro la diferencia entre una Base de Datos (BD) y un SGBD, pues la BD es el "contenedor" donde residen todos los datos y el SGBD es el que se encarga de manipular la información contenida en las BD's a través de operaciones de lectura y escritura de la información de las mismas [6].

2.4.1 Modelo relacional de datos

El modelo relacional fue propuesto por E.F. Codd en los laboratorios de IBM en California en 1970. Se trata de un modelo lógico, establece una estructura sobre los datos, aunque posteriormente éstos puedan ser almacenados de múltiples formas para

aprovechar características físicas concretas de la máquina sobre la que se implante la base de datos realmente. [7]

Sustituyó a los modelos más populares del momento: el jerárquico y el de red. Facilidad de uso y flexibilidad. Constituye la base para el sistema manejador de bd. Relacionales (RDBMS).

Es el más usado y extendido en el mundo de las BD's, la representación gráfica de este modelo sería una "tabla", compuesta de filas y columnas, donde las filas serían los registros de datos y las columnas los campos.

Codd propuso 12 reglas para el modelo relacional:

1. **Regla de Información:** Toda la información solo se presenta tablas.
2. **Regla de acceso garantizado:** Se accede a los campos sólo por nombre de columna y valor de llave candidata.
3. **Manejo sistemático de valores nulos:** Se debe disponer de una representación de valores desconocidos y no aplicables diferente de los valores normales.
4. **Catálogo activo en línea basado en el modelo relacional:** Debe estar a disposición de los usuarios con el mismo lenguaje de consulta que la base de datos.
5. **Sublenguaje de datos completo:**
 - Sintaxis lineal
 - Utilización interactiva y mediante lenguajes de programación
 - Definición de datos, manipulación completa de datos,
 - Restricciones de seguridad, integridad y manejo de transacciones
6. **Actualización de vistas:** Se deben poder actualizar todas las vistas que en teoría se puedan actualizar
7. **Inserción, modificación y borrado de alto nivel:** Se debe modificar, insertar y borrar todo un conjunto de tuplas a la vez
8. **Independencia física de los datos:** El acceso lógico a los datos debe mantenerse incluso cuando cambien los métodos de acceso o la forma de almacenamiento.

9. **Independencia lógica de los datos:** Los programas de aplicación no deben verse afectados por cambios realizados en las tablas que estén permitidos.
10. **Independencia de integridad:** La integridad no forma parte de los programas de aplicación sino del esquema conceptual
11. **Independencia a la distribución:** Debe existir un sublenguaje de datos que pueda soportar bases de datos distribuidas sin alterar los programas de aplicación cuando se distribuyan los datos por primera vez o se redistribuyan estos posteriormente.
12. **No subversión:** Si el sistema ofrece una interfaz de acceso a bajo nivel, dicho acceso no se podrá usar para salvar restricciones de integridad o seguridad

El modelo Relacional tiene las siguientes ventajas:

1. **Independencia lógica y física de los datos,** de las aplicaciones.
2. **Redundancia mínima,** ya que la base funciona como repositorio común de datos para distintas aplicaciones.
3. **Acceso concurrente** por parte de múltiples usuarios.
4. **Distribución espacial de los datos,** ya que la independencia lógica y física facilita las bases de datos distribuidas. Los datos pueden encontrarse en otra habitación, otro edificio e incluso otro país. El usuario no tiene por qué preocuparse de la localización espacial de los datos a los que accede.
5. **Integridad de los datos,** ya que se puede adoptar medidas de seguridad que impidan que se introduzcan datos erróneos. Esto puede suceder tanto por motivos físicos (defectos de hardware, actualización incompleta debido a causas externas), como de operación (introducción de datos incoherentes).
6. **Consultas complejas con más de una condición.**
7. **Seguridad de acceso y auditoria,** posibilita otorgar derecho de acceso diferenciado a personas y organismos. El sistema de auditoría mantiene el control de acceso a la base de datos, con el objeto de saber qué o quién realizó una determinada modificación y en qué momento.
8. **Respaldo y recuperación,** que posibilita al sistema recuperar su estado en un momento previo a la pérdida de datos.
9. **Acceso a través de lenguajes de programación estándar,** brinda la posibilidad ya mencionada de acceder a los datos mediante lenguajes de programación ajenos al sistema.

2.4.2 Relaciones entre tablas

Sean las tablas A y B, Existen tres formas en la que se pueden relacionar Figuras 2.3, 2.4 y 2.5):

Uno a uno: Cuando cada unidad de la tabla A está relacionada solo con una unidad de la tabla B

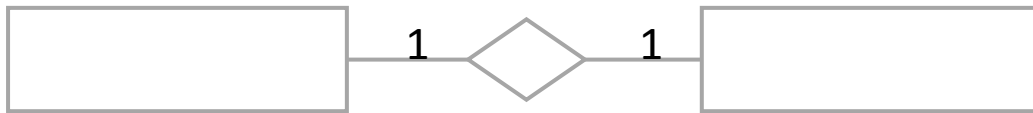


Fig. 2.3 Relación uno a uno

Uno a muchos: Si una unidad cualquiera de la tabla A está relacionada con varias de la tabla B, pero cada unidad de B está en relación con sólo una unidad de A.

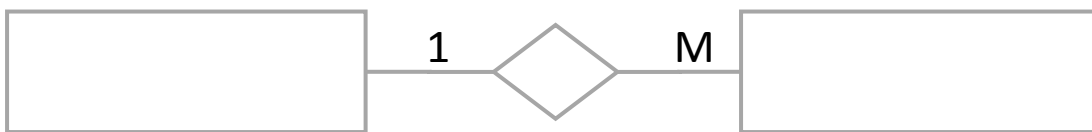


Fig. 2.4 Relación uno a muchos

Muchos a muchos: Si una unidad cualquiera de la tabla A, está relacionada con varias de la tabla B y una unidad cualquiera de B, está en relación con varias de la tabla A.

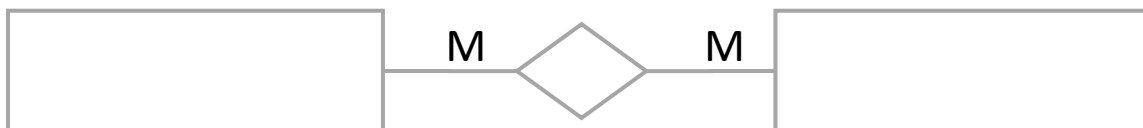


Fig. 2.5 Relación muchos a muchos

2.4.3 Modelado Entidad-Relación (ER)

Formalmente fue creado por Peter Chen en 1976 es una herramienta que nos sirve para representar la información del mundo real a nivel conceptual. Las características de este

modelado son: 1) Notación gráfica, 2) Semántica clara, 3) Es fácil de entender hasta por quienes no son especialistas y 4) Es independiente de cualquier manejador de datos [8].

Este modelo está basado en la percepción que tenemos del mundo real, se identifica por el conjunto de objetos básicos llamados entidades y relaciones, para representar esta percepción del mundo real con este modelo se utilizan gráficos llamados Diagramas Entidad Relación. (Ver anexo A.6).

Entidades: Las definimos como cualquier objeto real o abstracto que existe en un contexto determinado, que se diferencian entre sí, y del cual deseamos guardar información o realizar alguna operación. Se pueden clasificar en entidades físicas o reales y entidades abstractas o conceptuales. Estas a su vez tienen atributos.

Relación: Es un vínculo o correspondencia, que nos permite definir una dependencia entre varias entidades, es decir, nos permite exigir que varias entidades compartan ciertos atributos de forma indispensable. Según su **cardinalidad** se puede dividir en: uno a uno, uno a varios, varios a uno y varios a varios. Se define el **grado** según el número de entidades que participan en una relación. Al igual que la entidad las relaciones también pueden tener atributos.

Atributos: Son características o propiedades de las entidades, estos pueden tomar un valor en una instancia particular. Estos pueden ser simples o compuestos, univaluados o multivaluados, derivados o almacenados, nulos o claves

Claves: Es el atributo de una entidad, al que le aplicamos una restricción que lo distingue de los demás registros (no permitiendo que el atributo específico se repita en la entidad) o le aplica un vínculo (exactamente como comentábamos en las relaciones). Estos son los distintos tipos:

- a) **Superclave:** Aplica una clave o restricción a varios atributos de la entidad, para así asegurarse que en su conjunto no se repitan varias veces y así no poder entrar en dudas al querer identificar un registro.
- b) **Clave primaria:** Identifica a un solo atributo no permitiendo que se repita en la misma entidad.

- c) **Clave externa o clave foránea:** Debe estar estrictamente relacionado con la clave primaria de otra entidad, para así exigir que exista previamente ese clave.

2.4.4 Modelado ER-Extendido (ERE)

Este modelo incluye todos los conceptos del modelo ER, se ven nuevos conceptos como los de: **subclase, superclase, especialización, generalización y categorización**. Asociado con estos conceptos se encuentra el mecanismo de herencia en el atributo. Una de las desventajas que tiene el ERE es que no existe una terminología estándar para estos conceptos, por lo que mostraremos la más empleada hasta estos momentos. (*Ver anexo 7*).

Subclases, superclases: Se utilizan cuando se necesita explícitamente representar entidades que tienen varios subgrupos de entidades. La relación CLASE/SUBCLASE es a veces conocida como una relación. Una entidad no puede existir en una base de datos solo porque es miembro de una subclase, debe ser también miembro de la superclase. Sin embargo, no es necesario que toda entidad que pertenezca a una superclase sea miembro de alguna subclase.

La herencia en los atributos: Una entidad de una subclase representa la misma entidad de la superclase y posee atributos específicos que pertenecen a la subclase y atributos que son parte de la superclase. Toda entidad que es miembro de una subclase hereda todos los atributos de la superclase. Una subclase, con todos los atributos que hereda de la superclase, es un tipo de entidad.

Especialización: La especialización es el definir un conjunto de subclases de un tipo de entidad llamada la superclase. El conjunto de subclases que forma una especialización se define sobre la base de algunas características que lo distinguen de otras especializaciones dentro de la misma superclase. En resumen la especialización nos permite:

- Definir un conjunto de subclases de un tipo de entidad.
- Asociar atributos específicos a cada subclase.
- Establecer tipos de relación adicionales entre subclases y entre cada subclase y otros tipos de entidad u otras subclases.

Generalización: Se puede pensar en un proceso de reversa en el cual se suprimen las diferencias sobre varios tipos de entidad y se identifican sus rasgos comunes, y se

generalizan a una superclase del cual un tipo de entidad es una subclase en especial. Se entiende como generalización al proceso de definir un tipo de entidad generalizado a partir de tipos de entidades que se pueden agrupar. En sí la generalización se puede ver como el proceso inverso de la especialización.

La extensión de este modelo varía dependiendo de los puntos de vista de los autores, su contribución es añadir abstracción de generalización y especialización al modelo de Peter Chen, a continuación nombraremos algunos ejemplos de esta extensión.

De Teorey, Yang y Fry: Estos autores hacen énfasis en la abstracción de categorías en el modelado Entidad Relación y adicionan 2 tipos de objetos, las jerarquías de subconjuntos y jerarquías de generalización. Ellos dicen que las jerarquías de subconjuntos especifican subconjuntos con posibles intersecciones y las jerarquías de generalización especifican subconjuntos estrictamente disjuntos. (*Ver anexo 8*).

De Markowitz y Shoshani: Describen la generalización como la de Teorey, Yang y Fry, como un mecanismo de abstracción que permite ver un conjunto de conjuntos de entidades como un único conjunto de entidad genérico. Markowitz y Shoshani describen a la generalización como una relación transitiva. Un tipo-entidad especializado hereda los atributos de todos los tipos-entidad genéricos directos y transitivos, incluyendo el identificador del tipo-entidad. Estos atributos son llamados atributos heredados. Los tipos de entidad que no son débiles, y no son una especialización de otro tipo-entidad, son llamados tipo-entidad “independientes”. (*Ver anexo 9*)

ECER: Es llamado Entidad-Relación Conceptual Extendido desarrollado por el autor Czejdo, usa interfaces gráficas para formular consultas y actualizar. Este concepto tiene las ventajas del diagrama ER normalmente usados sólo para diseño de bases de datos y las aplica a las interfaces de usuario simples del tipo “apuntar y clicar”. Incluye operadores para actualizar, recuperar, calcular, etc. Czejdo define tres tipos de generalización/especificación: a) Involucra exactamente dos tipos-entidad, donde uno es un subconjunto del otro, b) involucra un tipo-entidad generalizado y varios tipos-entidad especializados y c) Este es como el tipo 2 excepto que los tipos especializados son disjuntos y están conectados en el diagrama a un círculo que contiene el símbolo más. (*Ver anexo 10*).

Hohenstein y Gogolla: Es definido en conexión con una proposición de cálculo para el modelo. Es muy parecido al de Teorey, además de estar basado en el de Chen, porque incluyen generalización. Presentan a la generalización y especialización como construcciones de tipos con tipos de entrada y tipos de salida. Los tipos de entrada son usados en la construcción de tipos de

salida, esto significa que todas las entidades de los tipos de entrada son agrupadas y distribuidas entre los tipos de salida. (Ver anexo 11)

ECR: El autor propone un modelo de entidad–categoría-relación (ECR). Una categoría es “un subconjunto de entidades de un tipo-entidad”. Un conjunto de entidades es llamada clase de objetos, sea éste tipo-entidad o una categoría. Las categorías usualmente comparten la mayoría de los atributos, pero no necesariamente todos. El modelo ECR usa notaciones en el diagrama similares a las de Czejdo. Las categorías son mostradas en el diagrama como hexágonos, conectados al tipo-entidad con un símbolo de subconjunto dibujado sobre la línea. Los atributos compartidos son conectados al tipo-entidad y los atributos de categoría no compartidos son conectados al hexágono. Las categorías son el foco de la integración de vistas. Las categorías son definidas para cada fuente de datos para la integración y son comparadas durante el proceso.

Los modelos extendidos anteriores están dirigidos al esquema de Bases de Datos Relacionales, sin embargo existen otras extensiones al Modelo Entidad Relación que están Orientados a Objetos, estos surgieron por los continuos adelantos en las bases de datos orientadas a objetos. Estos modelos incluyen características como las de generalización y herencia, el principio de ocultamiento de la información, tipos de datos abstractos y traspaso de mensajes. Estas son, solo por listarlas: Entidad Relación Orientada a Objetos (EROO) de Navathe y Pillalamarri , Entidad Relación con Comportamiento Integrado (ERCI) de Kappel y Schrefl, Entidad Relación Orientada a Objetos Mejorado por Gorman y Choobinech [9].

2.4.5 Lenguaje estructurado de datos (SQL)

Después de que Codd en 1970 propusiera el modelo relacional y un sublenguaje basado en el cálculo de predicados, asociado a este modelo, Donalk Chamberlin de IBM en 1974 realiza el primer intento de crear un lenguaje estructurado de datos, este lenguaje se llamó SEQUEL (Structured English QUery Lenguaje), que fue usado por el sistema experimental System R, desarrollado por IBM. Más tarde en 1979 Oracle lo introduce en un programa comercial, posteriormente por motivos legales adquirió el nombre de SQL. En 1986 el ANSI transformó el lenguaje SQL en estándar ISO, con esto se originó la primera versión de este lenguaje llamada SQL-86 o SQL1. Este lenguaje a tenido varias revisiones originándose de estas el SQL89, SQL92 y el actual SQL3 [10]s.

La ventaja de tener un lenguaje estándar para bases de datos relacionales es la fácil comunicación entre los sistemas que lo usan, además de ofrecer portabilidad y fácil

uso. Este lenguaje es declarativo y de alto nivel, esto significa que especifica qué es lo que se quiere hacer y no cómo conseguirlo.

SQL a su vez se clasifica en tres lenguajes (figura 2.6):

1. Lenguaje de Definición de Datos (**LDD**), contiene instrucciones que sirven para definir el esquema de la base de datos.
2. Lenguaje de Manipulación de Datos (**LMD**), son el conjunto de instrucciones que permiten la manipulación de datos de las tablas.
3. Lenguaje de Control de Datos (**LCD**), contiene instrucciones que permiten dar y quitar permisos a los usuarios de los datos. Este garantiza la integridad, atomicidad y seguridad de los datos. (Ver *apéndice A.1*)

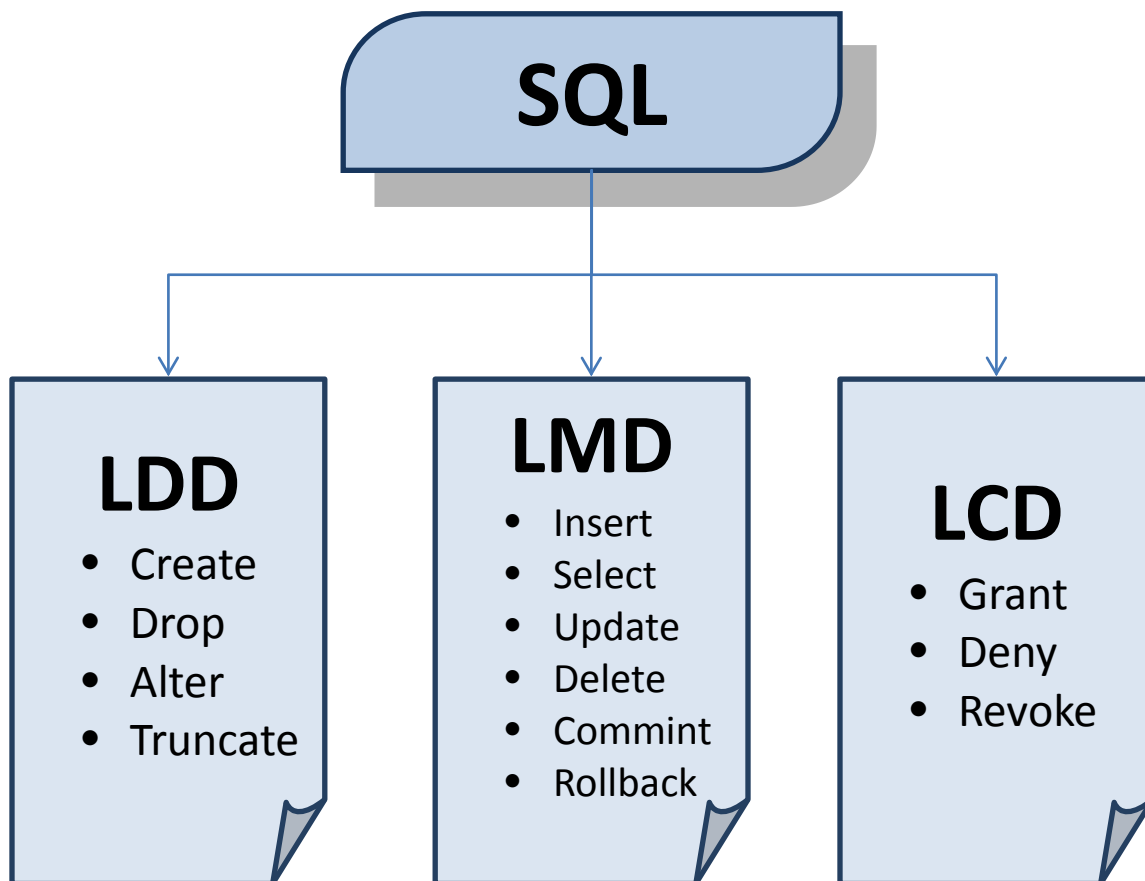


Figura 2.6 Clasificación del Lenguaje SQL

Cláusulas: Sirven para completar las sentencias, éstas especifican los datos sobre los cuales deben actuar las sentencias, además de proporcionar más detalles acerca de lo que se pretende consultar. (Ver apéndice A.2).

Operadores de Comparación: Sirven para hacer operaciones de comparación de columna a columna o de campo a campo, estas se pueden conectar con otras a través de operadores lógicos para crear expresiones más complejas. (Ver apéndice A.3).

Operadores Lógicos: Sirven para hacer operaciones de comparación de columna a columna o de campo a campo, estas se pueden conectar con otras a través de operadores lógicos. (Ver apéndice A.4).

Funciones Agregadas: SQL proporciona estas funciones agregadas que pueden tomar el nombre de la columna como un argumento y calcular el valor del operador agregado. (Ver apéndice A.5).

Normas de escritura: Como todos los lenguajes SQL también tiene restricciones de escritura para su mejor entendimiento.

- No distingue entre mayúsculas y minúsculas.
- Todas las instrucciones deben finalizar con punto y coma.
- Cualquier comando puede ser partido por espacios o saltos de línea antes de terminar la instrucción.
- Si fuera necesario se pueden tabular las líneas para facilitar su lectura
- Los comentarios se ponen entre `/* comentario */`, aunque solo algunos SDBD los soportan.

Tipos de datos: Entre los más importantes que SQL permite tenemos de tipo carácter, cadena, entero, real, fecha y booleano.

CAPÍTULO 3 ANÁLISIS DEL SISTEMA

En esta etapa recogeremos y analizaremos los requerimientos que tienen los usuarios del sistema, para esta recopilación nos auxiliaremos con entrevistas a los coordinadores y al titular de la Secretaría de Acción Electoral, observaremos el funcionamiento del área de interés y examinaremos los documentos que se deberán generar con el sistema.

3.1 OBJETIVOS DEL PROYECTO

El objetivo general del proyecto será automatizar el proceso de recolección de información de la estructura electoral, a través de un Sistema Web de tipo cliente-servidor.

A continuación enumeramos los objetivos específicos:

1. Facilitar el proceso de captura de datos personales de representantes.
2. Diseñar una BD para que concentre la información requerida de los representantes de todo el estado, clasificada por distritos electorales.
3. Verificar que no existan representantes repetidos en otras estructuras propias del partido.
4. Elaborar reportes semanales de avance de estructura electoral.
5. Imprimir solicitudes de registro de los representantes ante los órganos electorales.

3.2 PRODUCT BACKLOG

Para el desarrollo de este proyecto, se eligió SCRUM una metodología ágil, que permite la construcción de software con mayor rapidez que las metodologías de desarrollo de tradicionales. Uno de los principales componentes de SCRUM es el llamado Product Backlog o Pila del Producto, es una lista de requisitos de usuario que se generan de una visión inicial del producto a construir y es la antesala a la primera iteración o al inicio del primer sprint.

Por lo que dados los objetivos del proyecto que se dieron en el apartado anterior, tenemos que nuestra Pila del Producto la dividiremos en dos tipos de requerimientos: los funcionales y los no funcionales. En la tabla 3.1 se describen los requerimientos funcionales del sistema, que son los servicios que debe proporcionar el software o dicho de otra forma lo que debe hacer el mismo.

CÓDIGO	REQUERIMIENTO	PRIORIDAD
RF1.1	Facilitar la captura de información de los representantes, para ello debe contener alguna base de datos con la información de los electores.	ALTA
RF2.1	Guardar la información actual del representante y el cargo que se le ha asignado.	ALTA
RF3.1	Permitir cambios de representantes hasta una fecha límite.	ALTA
RF4.1	No permitir representantes repetidos, ni en la misma estructura electoral, ni con la de promoción al voto.	ALTA
RF5.1	Permitir la captura de algún representante solamente si este no ha sido insaculado (seleccionado para ser funcionario de casilla).	ALTA
RF6.1	Aceptar solo información de representantes que no hayan fungido en otros partidos políticos. Por lo que se requiere que el sistema contenga la base de datos de representantes de otros partidos al menos de la elección inmediata anterior.	MEDIA
RF7.1	Generar un reporte impreso, en todo momento, del avance de estructura electoral y quienes de estos ya están capacitados. Este reporte podrá ser estatal o distrital.	ALTA
RF8.1	Generar la impresión de la solicitud de registro del representante, con autorización previa del Secretario de Acción Electoral.	ALTA
RNF1.1	Facilitar el acceso al sistema desde cualquier computadora que esté conectada a internet.	ALTA
RNF2.1	Restringir el acceso al sistema, éste solo será permitido a las personas autorizadas para la captura, quienes solo tendrán la posibilidad de agregar datos, modificar y consultar la información correspondiente al distrito que tengan asignado, distinguiéndolas con un nombre de usuario y una contraseña.	ALTA
RNF3.1	Para el correcto funcionamiento del sistema, se requiere una computadora, con navegador de internet(i-explorer 5.0, Google Chrome 30.0, Mozilla), con velocidad mínima de transferencia de datos de 100Mb	MEDIA
RNF4.1	Proteger la información guardada en la base de datos, el sistema no debe permitir posibles intrusiones en la base de datos	ALTA

Tabla 3.1 Product Backlog

La descripción del requerimiento la encontramos en el campo REQUERIMIENTO, en el campo PRIORIDAD se asigna la preferencia de desarrollo que se le debe dar a un requerimiento y por último el campo CÓDIGO especifica el código del requerimiento y lo definimos en tres apartados como se muestra en la figura 3.1:

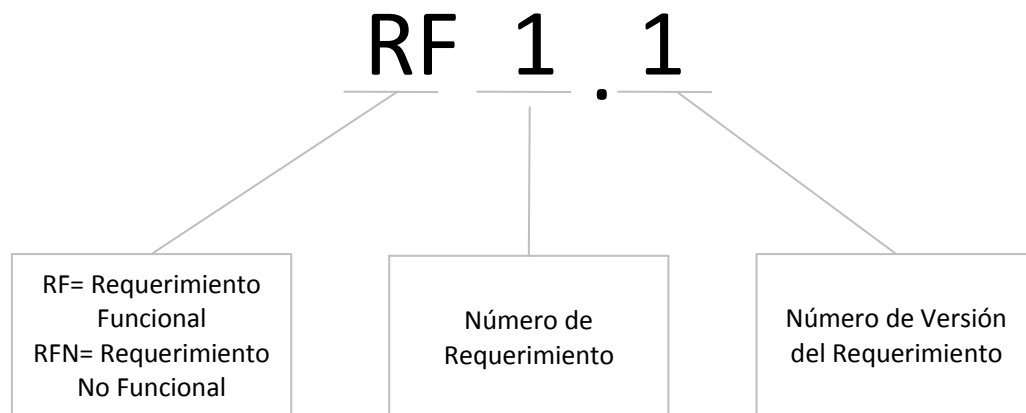


Fig. 3.1 Clasificación de los requerimientos

3.1 1^{er} SPRINT

Los creadores de Scrum, Ken Schwaber y Jeff Sutherland, se inspiraron en el juego de rugby para denominar así la metodología, pues ellos comparan la forma de trabajo de equipos altamente productivos y multidisciplinarios con la colaboración entre los jugadores de Rugby y su formación de Scrum, también llamada melé en español. El Sprint, en el rugby, se le llama a un "avance de balón" propuesto por el equipo, en la aplicación de esta metodología a nuestro proyecto, el Sprint se convierte en un "entregable" (producto entregable o prototipo del sistema que tiene partes funcionales). En Scrum, el Sprint también tiene su Backlog o Pila.

3.3.1 Backlog del 1^{er} Sprint

Antes de empezar el 1^{er} Sprint necesitamos tener la lista o pila de datos que se procurarán cumplir en esta etapa, estas son:

- ✓ Planeación: elaboraremos un esquema conceptual del sistema, a través de una representación gráfica utilizando algunas herramientas de UML.
- ✓ Diseño: realizaremos el diseño utilizando el esquema conceptual generado en la planeación, procurando de cumplir con el product backlog del sistema.
- ✓ Entrega de una primera etapa del sistema que cumpla con los requerimientos funcionales RF1.1, RF2.1, RF4.1, RF5.1, RF6.1, RNF1.1 y RNF2.1.

Comencemos pues con la planeación del sistema, para eso tendremos que hacer el análisis del sistema y las representaciones gráficas que se requieran. Como Scrum no es en sí una metodología de ingeniería de software, no define herramientas para la representación del análisis de sistemas, por lo que nos auxiliaremos con algunos diagramas de UML, para la parte del análisis de este proyecto [11].

3.3.1 Diagrama de contexto

Comenzaremos con el diagrama de contexto (figura 3.2), que nos da un panorama general de cómo debe ser el flujo de la información con base en la organización que tiene el Partido Político, en especial el área que esté encargada de los preparativos de las elecciones.

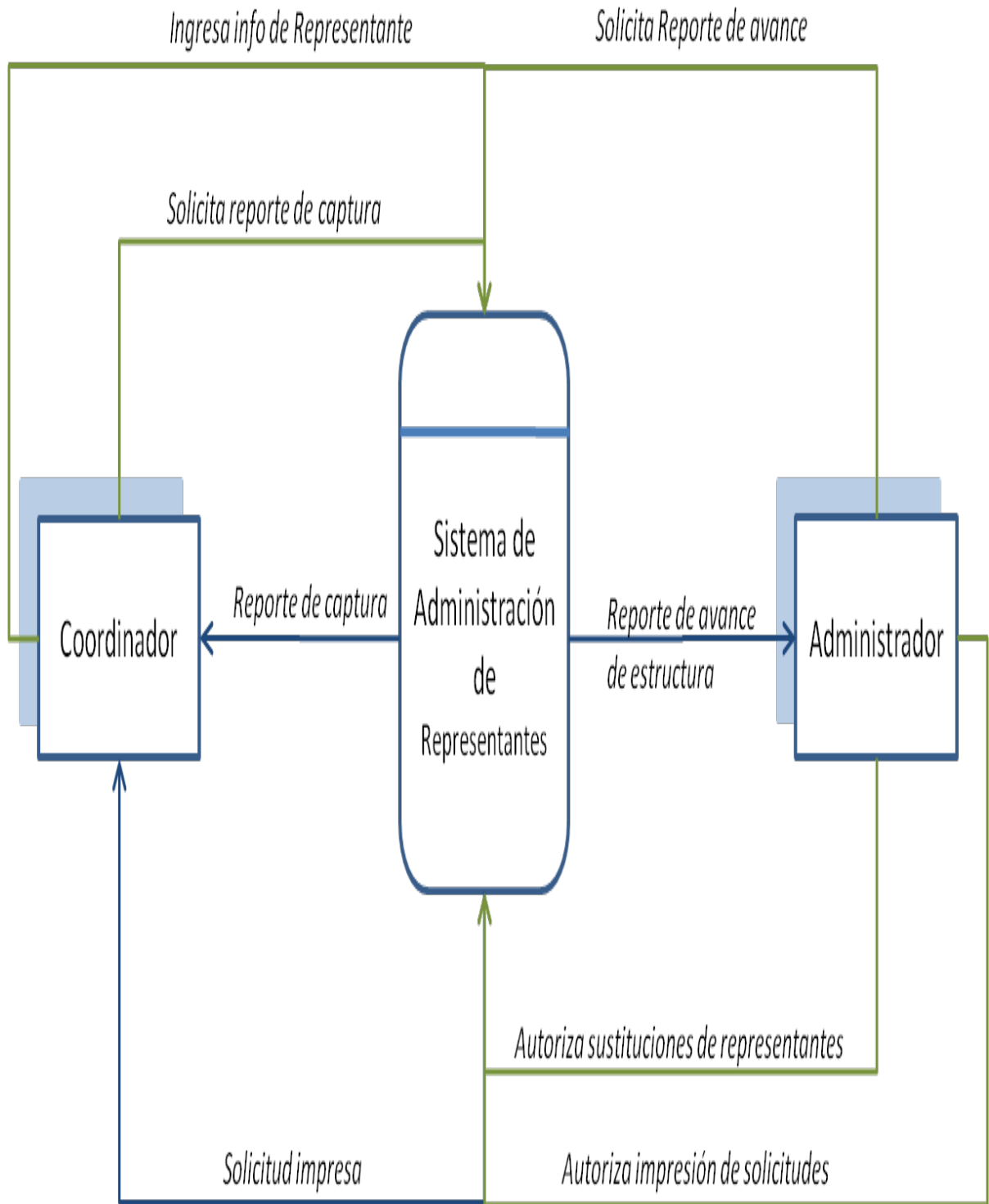


Fig. 3.2 Diagrama de Contexto del Sistema

3.3.2 Diagrama de casos de uso

Según los autores del libro de Análisis y Diseño de Sistemas, Kendall & Kendall, "los diagramas de caso de uso se están haciendo populares debido a su sencillez y carencia de detalles técnicos. Se utilizan para mostrar el alcance de un sistema, junto con las principales características del mismo y los actores que trabajan con esas características principales. Los usuarios que ven el sistema, pueden reaccionar a él y proveer retroalimentación".

Continuemos con el diagrama de casos de uso (figura 3.3), que nos servirá para representar cada historia de los usuarios, es decir representamos gráficamente qué es lo que hará el sistema, sin decir cómo lo va a hacer, para este caso tenemos dos actores o usuarios.

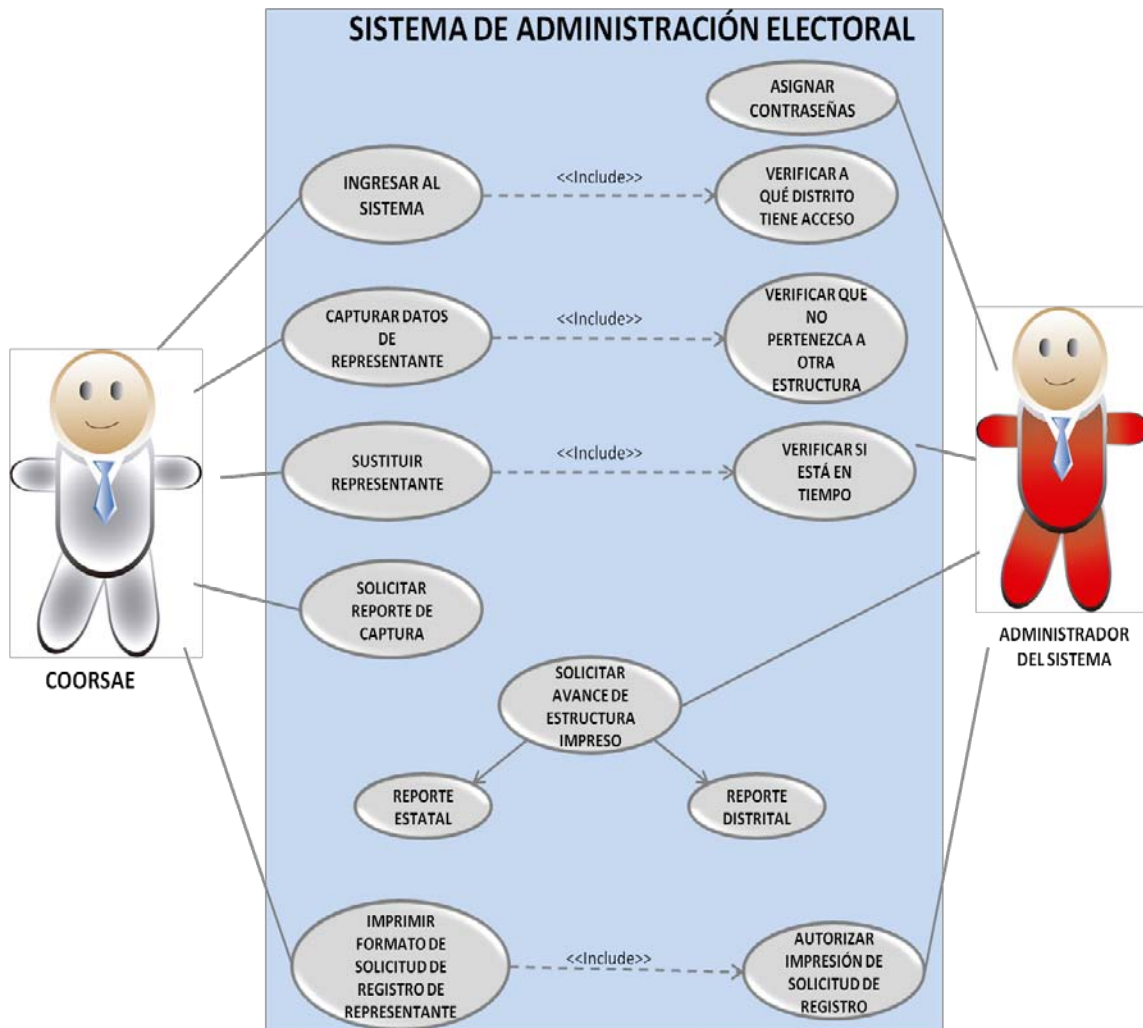


Fig. 3.3 Diagrama de Casos de uso del Sistema

3.3.3 Escenarios de casos de uso

En esta sección mostraremos los escenarios de casos de uso (tablas 3.2 a 3.8), que son las descripciones de cada caso de uso. Un escenario describe las transacciones representadas en los casos de uso. Posteriormente las figuras 3.4 y 3.5 corresponden al diagrama de clases y de secuencia respectivamente.

Nombre del Caso de Uso:	Asignar contraseñas	
Área:	Seguridad	
Actor(es):	Administrador	
Descripción	El administrador guardará en la BD las contraseñas de los Coordinadores	
Evento desencadenador:		
Pasos realizados (ruta principal)	Información de los pasos	
1. El Administrador registra en la BD la nueva contraseña del Coordinador	Contraseña de COORSAE	
Precondiciones:	No se le ha asignado contraseña al COORSAE	
Post condiciones:	Se guardó la nueva contraseña del Coordinador.	
Suposiciones:	N/A	
Requerimientos cumplidos:	RNF1.1, RNF2.1	
Prioridad:	Alta	
Riesgo:	Problemas con la conexión de la BD	

Tabla 3.2 Escenario de Caso de Uso

Nombre del Caso de Uso:	Ingresar al Sistema
Área:	Seguridad
Actor(es):	COORSAE
Descripción	El Coordinador de Acción Electoral ingresará al Sistema Web a través de una contraseña.
Evento desencadenador:	El COORSAE entra al sistema.
Pasos realizados (ruta principal)	Información de los pasos
1. El Coordinador de Acción Electoral ingresa al sistema mediante una página web.	Contraseña de COORSAE
2. Se lee la contraseña del COORSAE y se verifica si es correcta	Contraseña , Tabla de contraseñas
3. Se ingresa al módulo principal	
Precondiciones:	El COORSAE no necesita registrarse, pues se le asigna una contraseña de inicio.
Post condiciones:	El COORSAE tuvo éxito en el ingreso al sistema
Suposiciones:	La computadora en la que trabaja el COORSAE tiene conexión a internet y navegador Web y una contraseña de ingreso.
Requerimientos cumplidos:	RNF1.1, RNF2.1
Prioridad:	Alta
Riesgo:	Problemas con la conexión a internet

Tabla 3.3 Escenario de Caso de Uso

Nombre del Caso de Uso:	Capturar Representante
Área:	Control de Representantes
Actor(es):	COORSAE
Descripción	El Coordinador proporciona información del representante
Evento desencadenador:	Se muestra la interfaz de captura
Pasos realizados (ruta principal)	Información de los pasos
1. El Coordinador captura la clave de elector del representante.	Clave de elector
2. El Sistema busca los datos que corresponden a la clave de elector y verifica que el representante no esté registrado con algún otro cargo, que no pertenezca a promoción al voto o que haya sido insaculado.	Clave de elector, tabla de datos de electores, BD de representantes, tabla de datos de promoción al voto, tabla de datos de ciudadanos insaculados.
3. En caso de suceder que pertenezca a otra estructura el sistema muestra mensaje de que el representante que trata de registrar ya pertenece a otra estructura y muestra a cuál de todas.	Clave de elector, Nombre del representante.
4. El Sistema muestra la información correspondiente a la clave de elector	Clave de elector, Nombre, Domicilio, Municipio.
5. El Sistema pregunta al Coordinador, con respecto del representante, qué cargo se le va a asignar.	Cargo, Tipo y en su caso casillas.
6. El Sistema verifica si el cargo y las casillas no han sido asignadas, si lo han sido, muestra mensaje de notificación	Catálogo de casillas del estado
7. El Sistema pregunta al Coordinador si es correcta la información, en caso afirmativo, guarda información del representante en la BD.	
Precondiciones:	El coordinador debe tener las claves de elector de sus representantes y el cargo
Post condiciones:	Se guarda la información que corresponde al representante
Suposiciones:	N/A
Requerimientos cumplidos:	RF1.1, RF3.1, RF4.1, RF5.1, RF6.1
Prioridad:	Alta
Riesgo:	Problemas con la conexión a la BD e internet

Tabla 3.4 Escenario de Caso de Uso

Nombre del Caso de Uso:	Cambiar Representante
Área:	Control de Representantes
Actor(es):	COORSAE, Administrador
Descripción	El Coordinador modifica información de un representante registrado en el sistema
Evento desencadenador:	Se muestra la interfaz de modificación de datos de representante
Pasos realizados (ruta principal)	Información de los pasos
1. El Coordinador captura la clave de elector del representante.	Clave de elector
2. El Sistema busca la clave de elector en entre los representantes capturados	Clave de elector, BD de representantes
3. El Sistema muestra la información del representante	Clave de elector, Nombre, Domicilio, Municipio, Cargo, Tipo y en su caso casillas
4. El Coordinador modifica la información del representante	Cargo, Tipo y en su caso casillas.
6. El Sistema pregunta al Coordinador si es correcta la información, en caso afirmativo, guarda información del representante en la BD.	
Precondiciones:	El coordinador debe tener las claves de elector de sus representantes y el cargo
Post condiciones:	Se guarda la información que del representante
Suposiciones:	N/A
Requerimientos cumplidos:	RF3.1, RF4.1, RF5.1, RF6.1
Prioridad:	Alta
Riesgo:	Problemas con la conexión a la BD e internet

Tabla 3.5 Escenario de Caso de Uso

Nombre del Caso de Uso:	Solicitar Reporte de Captura	
Área:	Control de Representantes	
Actor(es):	COORSAE	
Descripción	El Coordinador solicita al sistema un reporte de representantes que ha capturado.	
Evento desencadenador:	Se muestra interfaz de reporte de representantes	
Pasos realizados (ruta principal)	Información de los pasos	
1. El Coordinador solicita reporte de captura de sus representantes a través de una opción en el menú principal.		
2. El Sistema muestra el número y la lista de representantes capturados, del distrito correspondiente al Coordinador.	Datos de los representantes del distrito, número total de representantes y número de capturados.	
Precondiciones:	Tener como mínimo un representante capturado	
Post condiciones:	Se muestre el reporte del distrito	
Suposiciones:	N/A	
Requerimientos cumplidos:	RF7.1	
Prioridad:	Alta	
Riesgo:	Problemas con la conexión a la BD e internet	

Tabla 3.6 Escenario de Caso de Uso

Nombre del Caso de Uso:	Solicitar Avance de Estructura impreso	
Área:	Control de Representantes	
Actor(es):	Administrador	
Descripción	El administrador solicita un reporte estadístico de representantes capturados.	
Evento desencadenador:	Impresión de reporte estadístico.	
Pasos realizados (ruta principal)	Información de los pasos	
1. El Administrador solicita reporte estadístico de captura de representantes en el estado o por cada uno de los distritos, a través de una opción en el menú principal.		
2. El Sistema muestra estadístico de representantes, (cuántos de cuántos), este reporte puede ser por cada uno de los distritos o estatal.	Total de representantes, total de representantes capturados.	
Precondiciones:	Tener como mínimo un representante capturado	
Post condiciones:	Se muestre el reporte estatal o distrital	
Suposiciones:	N/A	
Requerimientos cumplidos:	RF7.1	
Prioridad:	Alta	
Riesgo:	Problemas con la conexión a la BD e internet	

Tabla. 3.7 Escenario de Caso de Uso

Nombre del Caso de Uso:	Imprimir formato de solicitud de registro de representante
Área:	Control de Representantes
Actor(es):	COORSAE, Administrador
Descripción	El administrador autoriza la impresión de los formatos de solicitud de registro de los representantes.
Evento desencadenador:	Impresión de formato de solicitud de registro de representante
Pasos realizados (ruta principal)	Información de los pasos
1. El Administrador autoriza la impresión de formatos de determinado distrito.	Bandera de autorización
2. El Sistema activa opción de imprimir formatos de registro de determinado distrito	Bandera de autorización
3. El Coordinador imprime formatos a través de la opción de impresión de formatos.	Datos de los representantes del distrito, número total de representantes y número de capturados.
Precondiciones:	Captura del representantes al 100%del distrito correspondiente.
Post condiciones:	Impresión de formatos de registro
Suposiciones:	N/A
Requerimientos cumplidos:	RF8.1
Prioridad:	Alta
Riesgo:	Problemas con la conexión a la BD e internet, la impresora y los formatos preimpresos.

Tabla 3.8 Escenario de Caso de Uso

4.3.4 Diagrama de clases

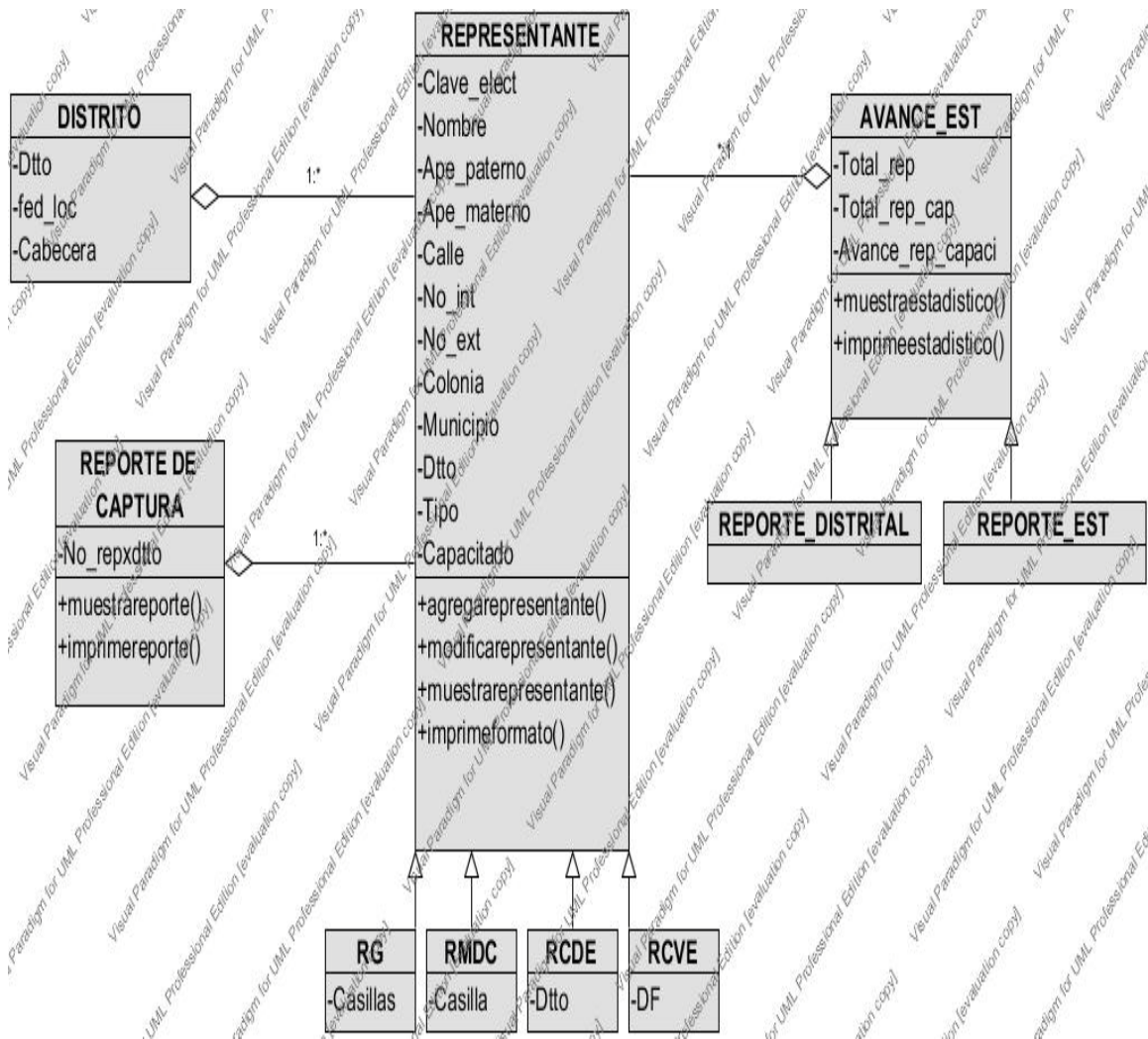


Fig. 3.4 Diagrama de Clases del Sistema

4.3.5 Diagrama de secuencia

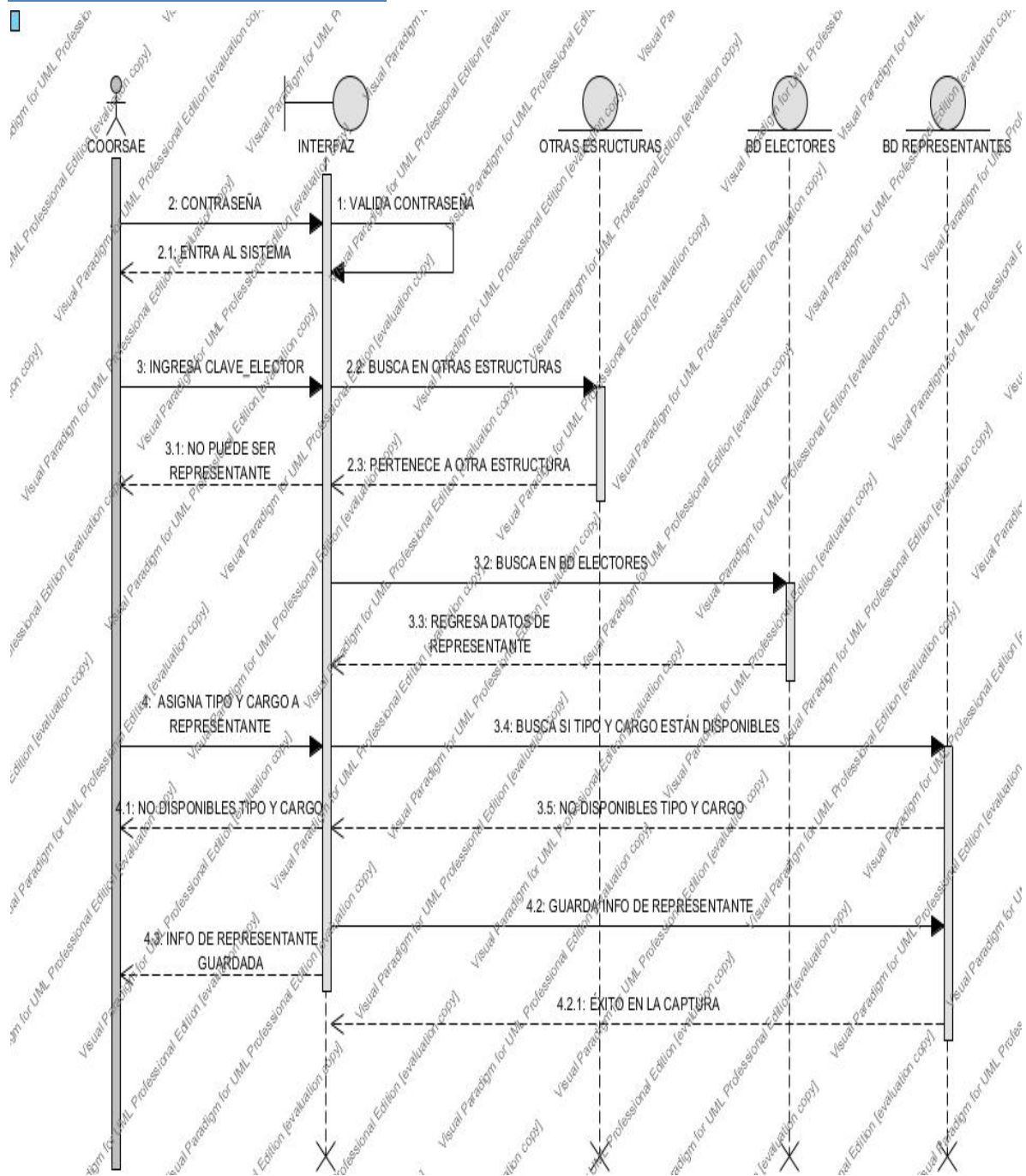


Fig. 3.5 Diagrama de Secuencia del Sistema

CAPÍTULO 4 DISEÑO DEL SISTEMA

Hasta ahora hemos hecho el análisis y planeación del sistema, en este capítulo veremos cómo realizar el Diseño del Sistema. Para una mejor planeación del Diseño del Sistema lo dividiremos en tres rubros: a) Diseño Arquitectónico del Sistema, b) Diseño de la Interfaz de Usuario y c) Diseño de la Base de Datos.

4.1 DISEÑO ARQUITECTÓNICO DEL SISTEMA

Uno de los requerimientos del Product Backlog es que el sistema sea de tipo web para que los coordinadores accedan a él en cualquier parte del estado a donde fueron asignados. Para poder lograr este objetivo utilizamos el modelo de red Cliente-Servidor, que por definición es un Sistema Distribuido en el que existen diferentes procesadores donde existen clientes que solicitan servicios y servidores que los proporcionan. Para adaptar este modelo de Cliente-Servidor al software en desarrollo, haremos uso de un servidor y varios clientes. El servidor deberá estar ubicado en la sede central del partido político y los clientes podrán ser cualquier computadora ubicada en diferentes partes del estado que tengan acceso a Internet.

Apache es un servidor de páginas web gratuito y de código abierto, que nos ayudará a adaptar este modelo de Cliente-Servidor, porque permitirá acceder a páginas web a los clientes, es decir, permitir a los coordinadores acceder al sistema. En la figura 4.1 mostramos el diseño arquitectónico con el que contará el sistema en desarrollo.

Arquitectura Cliente-Servidor

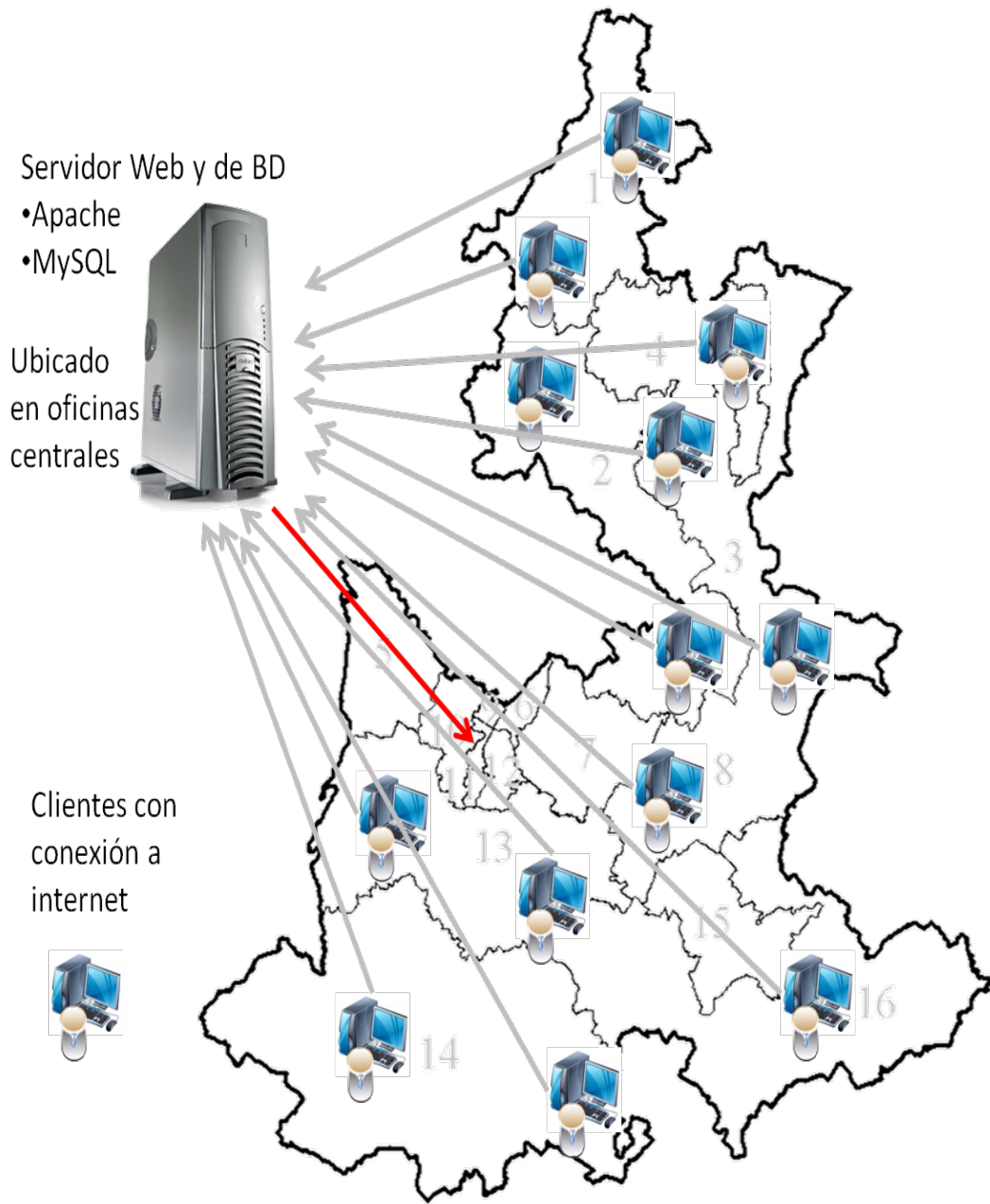


Fig 4.1 Arquitectura del Sistema de Representantes

4.2 DISEÑO DE LA BASE DE DATOS

Derivado del análisis de requerimientos observamos que se necesita crear una Base de Datos para el resguardo de la información que se utiliza en el sistema. Es importante representar los datos y las relaciones que existen entre ellos. Debemos entonces proporcionar un modelo de datos que soporte las transacciones requeridas por el sistema. Cuando se habla de diseño de base de datos se involucran tres etapas: el conceptual, el lógico y el físico.

4.2.1 Diseño Conceptual

En esta sección realizaremos el diseño conceptual de la Base de Datos requerida, utilizando la estrategia de arriba a abajo ya que se trata de una base de datos compleja. Comenzaremos por realizar un esquema de la información que se usará en el sistema en desarrollo, a través de un diagrama derivado del Modelado Entidad Relación (figura 4.2). Para lograr la elaboración de este diagrama hicimos la identificación de entidades de alto nivel y las fuimos refinando hasta encontrar entidades de bajo nivel de las cuales pudimos derivar los atributos, el dominio que tienen éstos, las relaciones que existen entre las entidades y algunas posibles claves primarias. Observamos que existe clasificación de entidades, por lo que el diagrama generado será derivado del Modelado Entidad Relación Extendido (ERE). Es importante mencionar que en las reuniones diarias del 1er. Sprint, el diagrama ERE sugerido tuvo el visto bueno de los usuarios del sistema, pues este fue generado de la información del Product Backlog y la observación de los procesos de la Secretaría que utilizará el sistema. En la siguiente figura mostramos el diagrama Entidad Relación Extendido (ERE) que representa el modelo de datos de la estructura de representantes de un partido político y sus características.

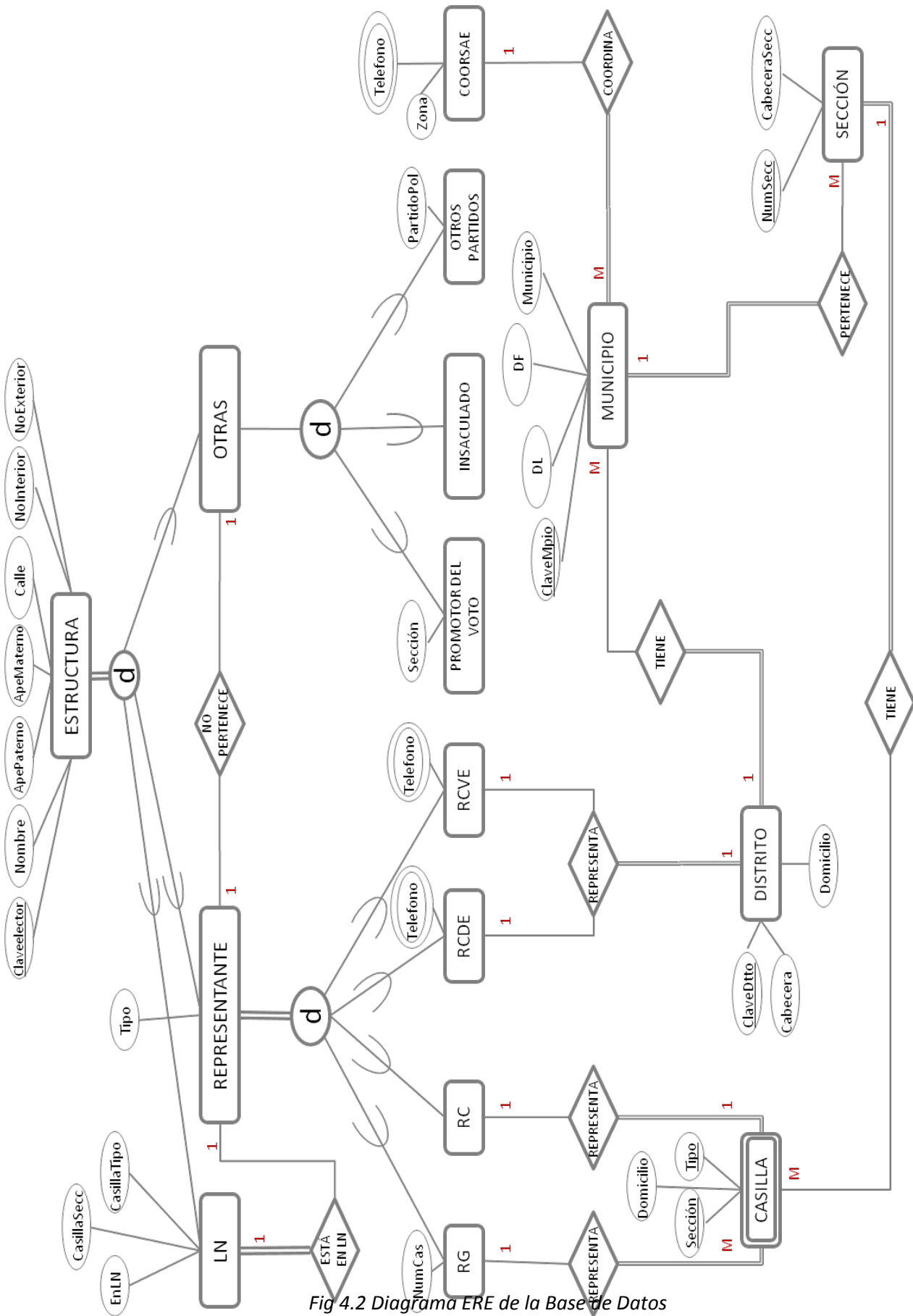


Fig 4.2 Diagrama ERE de la Base de Datos

4.2.2 Diseño Lógico

En esta segunda etapa convertiremos el diseño conceptual (diagrama ERE que elaboramos en la sección anterior), en un esquema lógico relacional, siguiendo el algoritmo de mapeo ER-tablas. En este proceso construiremos un esquema de la información independiente del SGBD que seleccionaremos más adelante en secciones subsecuentes.

4.2.2.1 Mapeo ER-Tablas

- ✓ **Primero buscaremos eliminar de nuestro esquema conceptual las estructuras de datos que los sistemas relacionales no modelan directamente. Para esto debemos cumplir con los pasos de la tabla 4.1:**

Paso		Resultado de la observación del diagrama ER
1	<i>Eliminar las relaciones de muchos a muchos</i>	No existen
2	<i>Eliminar las relaciones entre tres o más entidades</i>	No existen. Aunque aparentemente la relación "representa" entre RCDE, RCVE y Distrito, pareciera que se encuentra entre tres entidades, no es así, pues "representa" entre RCDE y Distrito se refiere a RCDE -> "representa Consejo Distrital" -> Distrito y así sucede con la relación entre RCVE -> "representa Comisión de Vigilancia -> Distrito.
3	<i>Eliminar las relaciones recursivas</i>	No existen
4	<i>Eliminar las relaciones con atributos</i>	No existen
5	<i>Eliminar los atributos multievaluados</i>	Existen 3: Teléfonos de Coordinador, de RCDE y RCVE. No los eliminamos, dejamos la solución en los siguientes pasos*.
6	<i>Revisar las relaciones de uno a uno</i>	No existen entidades que se dupliquen.
7	<i>Eliminar las relaciones redundantes</i>	No existen

Tabla 4.1. Secuencia de pasos del Algoritmo de mapeo

- ✓ **Segundo, derivado del análisis anterior concluimos que no tendremos que hacer alguna modificación a nuestro diagrama ERE. Debemos continuar con la elaboración de tablas para el esquema lógico global, como se muestra en los siguientes 6 pasos.**

1. **Crearemos una tabla para cada entidad fuerte**, con todos sus atributos simples y elegiremos la clave candidata que tenga menos atributos, que sus valores no cambien en el futuro, en caso de que sean varios atributos los que formen la clave candidata estos no pierdan su unicidad, además de que sean con el menor número de caracteres y de que esta sea fácil de usar para los usuarios.

LN	(<u>Claveelector</u> , Nombre , ApePaterno, ApeMaterno, Calle, NoInterior, NoExterior, Colonia, Municipio, CasillaSeccion, CasillaTipo,EnLN)
REPRESENTANTE	(<u>Claveelector</u> , Nombre , ApePaterno, ApeMaterno, Calle, NoInterior, NoExterior, Colonia, Municipio, Tipo, ClaveDtto, Telefono)
COORSAE	(<u>Claveelector</u> , Nombre , ApePaterno, ApeMaterno, Calle, NoInterior, NoExterior, Colonia, Municipio, Telefono, Zona)
PROM_VOTO	(<u>Claveelector</u> , Nombre , ApePaterno, ApeMaterno, Sección, Año)
INSACULADO	(<u>Claveelector</u> , Nombre , ApePaterno, ApeMaterno)
OTROSP	(<u>Claveelector</u> , Nombre , ApePaterno, ApeMaterno, PartidoPol, Año)
DISTRITO	(<u>ClaveDtto</u> , Cabecera, Domicilio)
MUNICIPIO	(<u>ClaveMpio</u> , Municipio, DF, DL)
SECCION	(<u>NumSecc</u> , CabeceraSecc)

2. En este paso debemos **crear una tabla para cada entidad débil**, solo encontramos la entidad CASILLA que para tener un correcto funcionamiento necesita de una llave compuesta y esta depende de la entidad SECCION por lo que la tabla que se necesita es la siguiente:

CASILLA (Seccion, Tipo, Domicilio)

3. Para cada **relación binaria uno a uno**, donde exista una relación entre tablas de padre a hijo incluiremos la clave primaria de la tabla padre a en la tabla hijo para que esta funcione como llave foránea. Las relaciones que cumplen lo anterior son las siguientes:

- | | | | | | |
|----|---------------|----|------------|----|----------|
| a) | REPRESENTANTE | -> | REPRESENTA | -> | DISTRITO |
| b) | REPRESENTANTE | -> | REPRESENTA | -> | CASILLA |
| c) | REPRESENTANTE | -> | REPRESENTA | -> | CASILLAS |

En la tabla 4.2 mostramos los atributos que se agregan

Tabla Padre		Tabla Hijo	Clave Primaria que se agrega al hijo
a)	DISTRITO	REPRESENTANTE	ClaveDtto
b)	CASILLA	REPRESENTANTE	<u>NumSecc, TipoCasilla</u>
c)	CASILLA	REPRESENTANTE	<u>NumSecc, TipoCasilla</u>

Tabla 4.2. Secuencia de pasos del Algoritmo de mapeo

4. Para cada **relación binaria uno a muchos**, donde exista una relación entre tablas de padre a hijo incluiremos la clave primaria de la tabla padre a en la tabla hijo para que esta funcione como llave foránea. Las relaciones que cumplen lo anterior son las siguientes:

a)	DISTRITO	->	TIENE	->	MUNICIPIO
b)	MUNICIPIO	->	TIENE	->	SECCIÓN
c)	CASILLA	->	PERTENECE	->	SECCIÓN
d)	COORSAE	->	COORDINA	->	MUNICIPIO

En la tabla 4.3 mostramos los atributos que se agregan

Tabla Padre		Tabla Hijo	Clave Primaria que se agrega al hijo
a)	DISTRITO	MUNICIPIO	ClaveDtto
b)	MUNICIPIO	SECCIÓN	ClaveMpio
c)	SECCIÓN	CASILLA	NumSecc
d)	COORSAE	MUNICIPIO	Claveelector

Tabla 4.3 Secuencia de pasos del Algoritmo de mapeo

5. Analizando nuestro esquema ER visualizamos que no existen **relaciones binarias de muchos a muchos**, por lo que continuamos con el siguiente paso.
6. En este punto daremos solución al * de la primera tabla al inicio de esta sección, de los **atributos multievaluados**, tenemos 3 atributos que se refieren a los teléfonos de los RCDE, RCVE y COORSAE, para quitar estos atributos multievaluados de cada tabla, crearemos una tabla exclusiva para los teléfonos, como se muestra a continuación:

TELÉFONOS (Claveelector, Teléfono, TipoTel)

Hasta este momento hemos terminado el mapeo y el esquema relacional resultante lo mostramos en la figura 4.3:

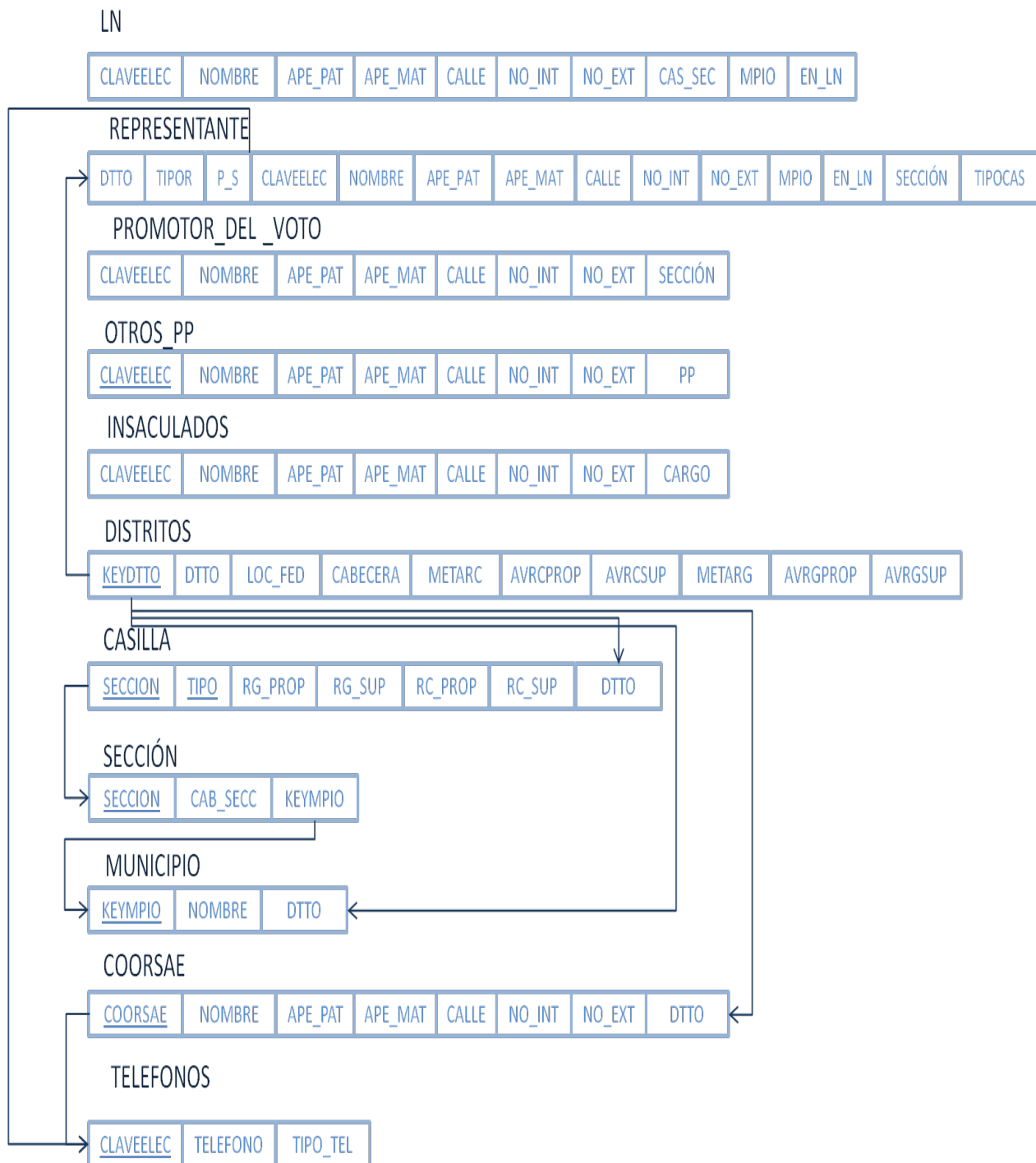


Fig. 4.3 Esquema Relacional de la BD

Para concluir con la etapa de diseño lógico de nuestra Base de Datos debemos aplicarle la Normalización.

4.2.2.2 Normalización

Utilizaremos una técnica para comprobar la validez del esquema lógico relacional que hemos generado hasta ahora, esta técnica se llama Normalización y nos ayudará a garantizar que las tablas obtenidas no tengan datos redundantes. Comencemos entonces con el proceso de normalización de la Base de datos.

Primera Forma Normal (1FN)

Para que una tabla esté en 1FN no debe tener **atributos multivaluados**, observamos que nuestro modelo de datos relacional nos exigió en el mapeo eliminar estos atributos multivaluados y se creó la tabla teléfonos. Por lo que todas las tablas de la BD cumplen esta forma.

Segunda Forma Normal (2FN)

Se dice que una tabla está en 2FN si cumple dos condiciones: a) que esté en 1FN y b) que todos los campos de la tabla dependan funcionalmente de las claves candidatas.

TABLA	DEPENDENCIA FUNCIONAL
LN	<u>Claveelector</u> → Nombre, Ape_Pat, Ape_Mat, Calle, No_Int, No_Ext, Colonia, Mpio, CasillaSeccion, CasillaTipo, EnLN
REPRESENTANTE	<u>Claveelector</u> → Nombre, Ape_Pat, Ape_Mat, Calle, No_Int, No_Ext, Colonia, Mpio, Tipo, Dtto, FedLoc, ClaveDtto
COORDINADOR	<u>Coorsae</u> → Nombre, Ape_Pat, Ape_Mat, Calle, No_Int, No_Ext, Colonia, Mpio,
PROM_VOTO	<u>Claveelector</u> → Nombre, Ape_Pat, Ape_Mat, Sección
INSACULADO	<u>Claveelector</u> → Nombre, Ape_Pat, Ape_Mat, Calle, No_Int, No_Ext, Colonia, Mpio, cargo
OTROSPP	<u>Claveelector</u> → Nombre, Ape_Pat, Ape_Mat, Calle, No_Int, No_Ext, Colonia, Mpio, PP
DISTRITO	<u>ClaveDtto</u> → Cabecera, Loc_Fed, metaRC, avanRCprop, avanRCsup, metaRG, avanRGprop, avanRGsup,
MUNICIPIO	<u>ClaveMpio</u> → Dtto, Nombre
SECCION	<u>NumSecc</u> → CabeceraSecc
CASILLA	<u>Seccion, Tipo</u> → Domicilio
TELÉFONOS	<u>Claveelector</u> → Teléfono, TipoTel

Tabla 4.4 Dependencias funcionales de las tablas de la BD

Todas las tablas (tabla 4.4) tienen dependencias funcionales totales y además las claves candidatas son unitarias a excepción de la tabla casillas que tiene una llave

compuesta por Sección y Tipo, pero Domicilio depende funcionalmente de la llave compuesta, por lo se cumple la 2FN en todas las tablas.

Tercera Forma Normal (3FN)

Para cumplir esta forma las tablas deben de estar en 2FN y además no tener dependencias transitivas. Todas las tablas cumplen con esta forma normal, ya que no existen dependencias transitivas.

Forma Normal de Boyce-Codd (FNBC)

La definición de esta forma dice que para que una tabla se encuentre en forma normal debe cumplir estar en 3FN y además que todo implicante en una dependencia funcional sea una llave candidata. Derivado la tabla anterior de dependencias funcionales afirmamos que todos los implicantes terminaron siendo llaves candidatas y como también cumplen la 3FN, también se cumple FNBC.

Este proceso fue totalmente iterativo, pues en cada reunión diaria del 1er. Sprint, fuimos refinando continuamente el diseño lógico. Al término de estos pasos hemos finalizado el proceso de normalización y con esto también el Diseño Lógico de la Base Datos.

4.2.3 Diseño Físico

El diseño físico está muy relacionado con el manejador de base de datos que se haya elegido, en este caso elegimos MySQL, por ser un Sistema Gestor de Bases de Datos Relacional, por lo que se adapta totalmente al modelo de datos relacional que elegimos en el diseño lógico. MySQL cuenta con un servidor multiusuario, rápido y robusto, que puede ejecutar instrucciones en paralelo de los usuarios que estén dentro de la red local o en internet.

4.3 DISEÑO DE LA APLICACIÓN

En esta etapa diseñaremos los programas de aplicación que harán uso de la Base de Datos y procesarán la información, así como el diseño de las interfaces de usuario, procurando que estas sean fáciles de aprender y de usar.

4.3.1 Diseño de la Interfaz de Usuario

En esta sección diseñaremos el aspecto que tendrá la interfaz de usuario. Debemos tomar en cuenta que los requerimientos mínimos que una interfaz de usuario necesita son: facilidad de entendimiento, ser atractivo a la vista, facilidad de captura y corrección apropiada que no hiera la sensibilidad del usuario.

Siguiendo estos mínimos requerimientos y tomando en cuenta que el sistema podría ser implementado para cualquiera de los 7 partidos políticos locales, usaremos como colores principales de la interfaz de usuario el rojo, el blanco y el gris, pues este sistema podría ser incrustado en las páginas propias de cada partido. Tratamos de procurar con el color blanco facilidad visual para los coordinadores en los prolongados tiempos de captura. Con el color rojo impacto y elegancia, con el color gris dar unos toques de seriedad.

En la sección 4.1 hablamos de la arquitectura Cliente-Servidor que debe tener el sistema, para cumplir con uno de los objetivos del Product Backlog, hicimos uso de Apache como Servidor de Páginas Web, para la creación de la página web haremos uso de HTML y para hacer un diseño gráfico adecuado a nuestras necesidades usaremos la tecnología CSS.

En el requerimiento no funcional RNF2.1 se pide restringir el sistema, pues solo deben tener acceso los coordinadores con un nombre de usuario y una clave, por lo que el sistema proporciona una interfaz de ingreso como se muestra en la siguiente figura.

Bienvenido al Sistema de Registro de Representantes,
por favor escribe tu Login y contraseña:



The image shows a simple login form with a white background and a thin border. It contains two text input fields: the first is labeled 'Login:' and the second is labeled 'Contraseña:'. Below these fields is a small rectangular button with the text 'Entrar' inside.

Fig. 4.4 Interfaz de Acceso al Sistema de Representantes

Para el diseño del logo, que será provisional porque el sistema puede adaptarse a cualquier otro partido político y para la elaboración de las imágenes que se muestran en los menús y submenús, usamos Adobe Illustrator CS5. La interfaz principal (figura 4.5) se

divide en 4 áreas principales: 1) El espacio de nombre del sistema y el logo que se encuentra en la parte superior, 2) El menú principal, que se encuentra del lado derecho justo abajo del nombre del sistema, 3) El submenú, ubicado debajo del logo y 4) El área de trabajo en la que se muestran los diferentes formularios o informes, esta depende totalmente de qué menú o submenú esté activo.



Fig. 4.5 Diseño de la Interfaz

CAPÍTULO 5 IMPLEMENTACIÓN DEL SISTEMA

En este capítulo veremos la implementación del sistema, debido a que este es un sistema que usa necesariamente una BD para su funcionamiento, tenemos que dividir la implementación en dos vertientes la implementación de la BD y la implementación programas de aplicación.

5.1 IMPLEMENTACIÓN DE LA BASE DE DATOS

La implementación de la BD se realiza mediante sentencias del lenguaje de definición de datos (LDD) del SGBD que elegimos, en este caso fue MYSQL. PhpMyAdmin es una interfaz gráfica de MYSQL, que nos facilitó la creación de la Base de Datos, en la figura 5.1 mostramos la base de datos

Paso 1. Creando la Base de Datos llamada Sistema Electoral

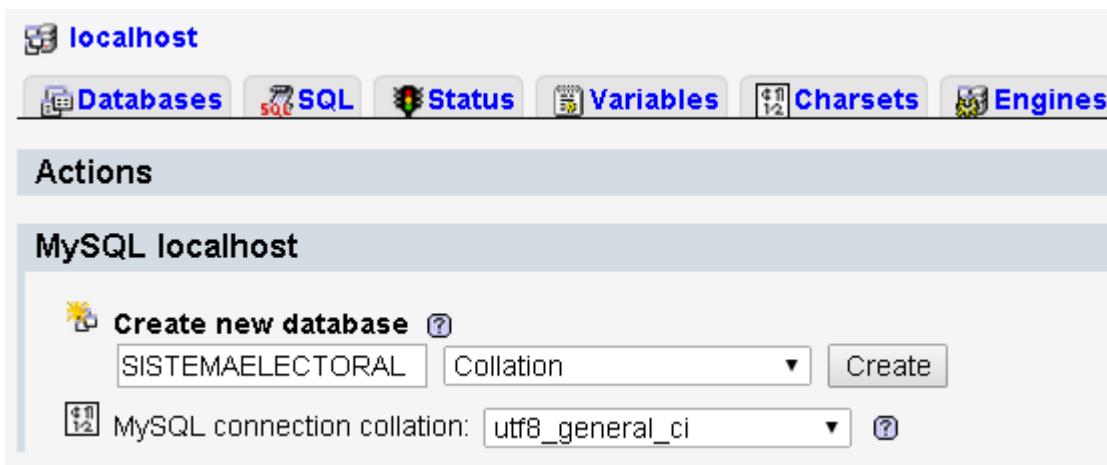


Fig. 5.1 Creación de la BD SistemaElectoral

Paso2. Creación de las tablas que integran la BD Sistema Electoral (figuras 5.2 a 5.4)

+ Routines

Print view Data Dictionary

Create new table on database **sae**

Name: REPRESENTANTES Number of fields: 13

i 1 May be approximate. See [FAQ 3.11](#)

Fig. 5.2 Creación de la Tabla REPRESENTANTES

localhost > sae > REPRESENTANTES

Field	Type	Length/Values1	Default?	Collation	Attributes	Null	Index	A I	Comments
Clavelector	VARCHAR	18	None			<input type="checkbox"/>	---	<input type="checkbox"/>	
Nombre	VARCHAR	50	None			<input type="checkbox"/>	---	<input type="checkbox"/>	
Ape_pat	VARCHAR	50	None			<input type="checkbox"/>	---	<input type="checkbox"/>	
Ape_mat	VARCHAR	50	None			<input type="checkbox"/>	---	<input type="checkbox"/>	
Calle	VARCHAR	100	None			<input type="checkbox"/>	---	<input type="checkbox"/>	
No_ext	VARCHAR	10	None			<input type="checkbox"/>	---	<input type="checkbox"/>	
No_int	VARCHAR	10	None			<input type="checkbox"/>	---	<input type="checkbox"/>	
Colonia	VARCHAR	50	None			<input type="checkbox"/>	---	<input type="checkbox"/>	
Mpio	VARCHAR	50	None			<input type="checkbox"/>	---	<input type="checkbox"/>	
Tipo	VARCHAR	20	None			<input type="checkbox"/>	---	<input type="checkbox"/>	
Dtto	INT	2	None			<input type="checkbox"/>	---	<input type="checkbox"/>	
FedLoc	INT	2	None			<input type="checkbox"/>	---	<input type="checkbox"/>	
ClaveDtto	VARCHAR	10	None			<input type="checkbox"/>	---	<input type="checkbox"/>	

Table comments: Storage Engine: InnoDB Collation:

Fig. 5.3 Creación de la Tabla REPRESENTANTES

Table	Action	Records	Type	Collation	Size	Overhead
casilla		6,912	MyISAM	utf8_general_ci	136.4 KiB	-
coordinador		2	InnoDB	latin1_swedish_ci	16.0 KiB	-
distrito		16	MyISAM	utf8_general_ci	2.2 KiB	80 B
insaculado		0	InnoDB	latin1_swedish_ci	16.0 KiB	-
ln		27	InnoDB	latin1_swedish_ci	16.0 KiB	-
municipio		220	MyISAM	utf8_general_ci	13.6 KiB	-
otrospp		0	InnoDB	latin1_swedish_ci	16.0 KiB	-
prom_voto		0	InnoDB	latin1_swedish_ci	16.0 KiB	-
rc		11	InnoDB	latin1_swedish_ci	16.0 KiB	-
rcde		1	InnoDB	latin1_swedish_ci	16.0 KiB	-
rcve		0	InnoDB	latin1_swedish_ci	16.0 KiB	-
rg		4	InnoDB	latin1_swedish_ci	16.0 KiB	-
secciones		2,619	MyISAM	utf8_general_ci	80.4 KiB	-
telefonos		0	InnoDB	latin1_swedish_ci	16.0 KiB	-
14 table(s)	Sum	9,812	InnoDB	latin1_swedish_ci	392.6 KiB	80 B

Check All / Uncheck All / Check tables having overhead With selected: ▾

Fig. 5.4 Vista de todas las tablas que integran la Base de Datos

5.2 IMPLEMENTACIÓN DE PROGRAMAS DE APLICACIÓN

En esta sección está dedicada a la implementación de los programas de aplicación en los que se incrustan las transacciones sobre la base de datos, estas transacciones se implementaron usando el lenguaje de manipulación de datos (LMD) de MYSQL.

Para la implementación de los programas de aplicación utilizamos el lenguaje de cuarta generación llamado PHP, que es un lenguaje de scripting que puede ser insertado dentro de páginas HTML y que nos permitirá escribir páginas con mayor rapidez. Otra de las características del PHP es que es un lenguaje del lado del servidor, es decir que se ejecuta del lado del servidor antes de que se envíe la página al cliente, es gratuito e independiente de la plataforma, por lo que nos permite la conexión con la BD, además de que PHP es compatible con todos los navegadores de internet. Con este lenguaje de programación se pueden realizar muchos tipos de aplicaciones web gracias a su enorme librería de funciones con la que está dotado. La librería de funciones cubre desde cálculos

5.3 SEGURIDAD

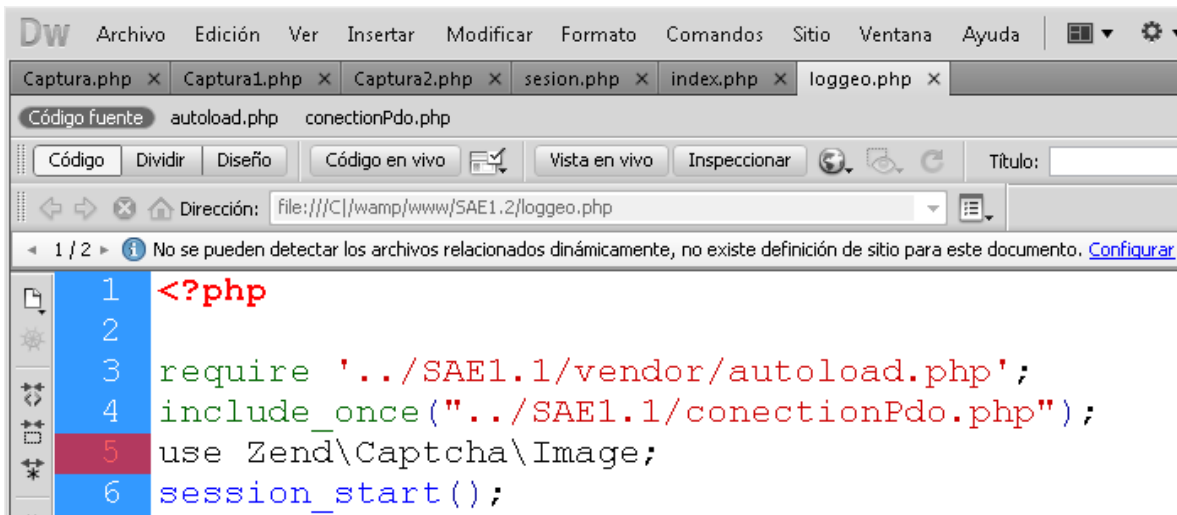
La seguridad en los sistemas es imprescindible, en particular en este sistema en desarrollo es de alta prioridad, uno de los requerimientos del sistema establece que éste esté protegido contra posibles ataques, por lo que implementaremos para el sistema integral las siguientes tácticas de seguridad.

5.3.1 Protección al Sistema

Implementaremos 2 mecanismos de seguridad para el ingreso al sistema estos son sesiones y generación de *captchas* para comprobar si es una persona la que ingresa al sistema y no un programa.

5.3.1.1 Sesiones

En combinación con el password y login, (requeridos a los usuarios), implementaremos con PHP el inicio y fin de sesión del usuario, para poder hacer operaciones sobre el distrito que le corresponde al usuario. Mediante PHP comprobaremos que los datos introducidos por el usuario en el formulario de login y password (figura 5.6) son válidos en la base de datos, y si es así iniciaremos una sesión guardándolos en el objeto `$_SESSION`. En cada página restringimos el acceso con este mecanismo.



The screenshot shows a web browser window with the address bar displaying `file:///C:/wamp/www/SAE1.2/loggeo.php`. The browser's developer tools are open, showing the source code of the page. The code is as follows:

```
1 <?php
2
3 require '../SAE1.1/vendor/autoload.php';
4 include_once("../SAE1.1/conectionPdo.php");
5 use Zend\Captcha\Image;
6 session_start();
```

```

if ($captcha->isValid($_POST['captcha'], $_POST)
    && isset($_POST['usuario']) && isset($_POST['contrasena']))
{
    $usuario = trim($_POST['usuario']);
    $contrasena = trim($_POST['contrasena']);
    $gbd->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
    foreach ($gbd->query("call loggeo('$usuario','$contrasena')") as $row)
    {
        $_SESSION['NombreCoord']=$row['Coor_nombre'];
        $_SESSION['PaternoCoord']=$row['Coor_paterno'];
        $_SESSION['MaternoCoord']=$row['Coor_materno'];
        header('Location:Captural.php');
    }
}
else {
    $_SESSION['usuario'] = $_POST['usuario'];
    $_SESSION['contrasena'] = $_POST['contrasena'];
    header('Location:index.php');
}
|
?>

```

Fig. 5.6 Inicio de sesión con PHP

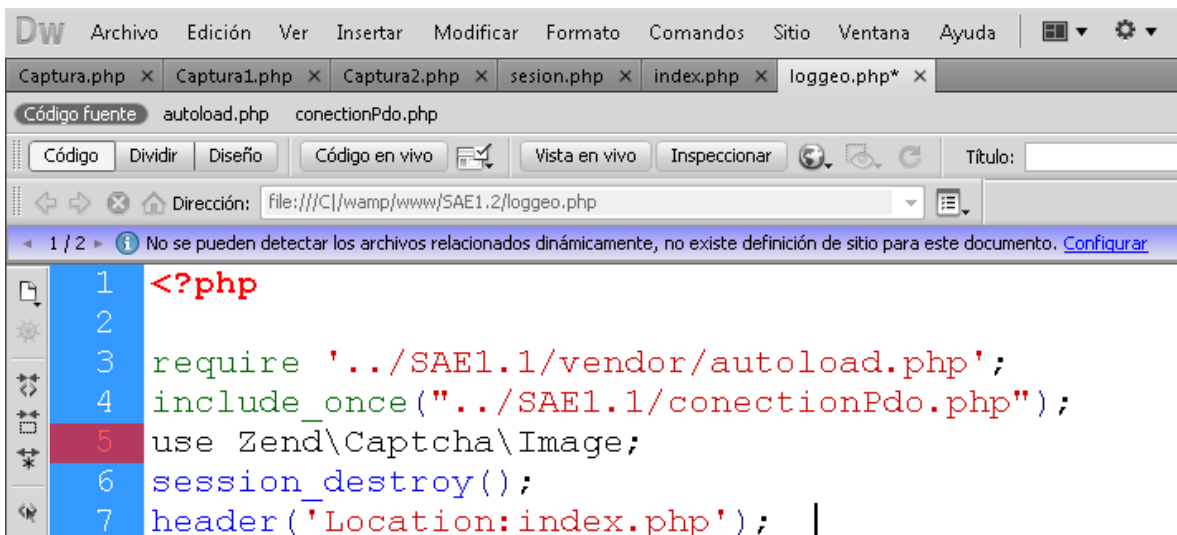


Fig. 5.7 Fin de sesión con PHP

5.3.1.2 Captchas

Otro mecanismo que utilizaremos antes de iniciar sesión y en conjunto con el formulario de password y nombre de usuario, es el de generar captchas para determinar cuándo el usuario es o no humano. En la figura 5.8 mostramos la interfaz gráfica de entrada al sistema con la generación de captchas aleatorios.



The screenshot shows a web interface with a dark red header containing navigation buttons: ENTRAR, CAPTURAR, ACTUALIZAR, BUSCAR, ESTADÍSTICAS, and SALIR. Below the header, there is a login form. On the left, there are three horizontal dashed lines. To the right, the text "Por favor ingresa los siguientes datos" is followed by two input fields labeled "USUARIO" and "CONTRASEÑA". Below these fields, the text "Introduce el código:" is followed by a captcha image showing the characters "54uehl" with a background of black dots and lines. A text input field is positioned below the captcha, and a button labeled "Ingresar" is at the bottom.

Fig. 5.8 Interfaz de entrada al Sistema con captcha.

```
<?php
require '../SAE1.1/vendor/autoload.php';
include_once("../SAE1.1/connectionPdo.php");
use Zend\Captcha\Image;
session_start();
$captcha = new Image(array(
    'name' => 'captcha',
    'wordLen' => 6,
    'timeout' => 300,
    'font' => 'fonts/arial.ttf',
    'fontSize' => 40,
    'imgDir' => './captchas/',
    'imgUrl' => 'captchas/',
    'dotNoiseLevel' => 30,
    'height' => 100
));
```

Fig. 5.9 Generación de Captchas con PHP (creación del objeto captcha)

En el código que mostramos en la figura 5.10 realizamos la autenticación, tomando en cuenta al nombre de usuario, la contraseña y el ingreso del código captcha. Con esto se garantiza la implementación del requerimiento RNF2.1.

```
if ($captcha->isValid($_POST['captcha'], $_POST)
    && isset($_POST['usuario']) && isset($_POST['contrasena']))
{
    $usuario = trim($_POST['usuario']);
    $contrasena = trim($_POST['contrasena']);
    $gdb->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
    foreach ($gdb->query("call loggeo('$usuario','$contrasena')") as $row)
    {
        $_SESSION['NombreCoord']=$row['Coor_nombre'];
        $_SESSION['PaternoCoord']=$row['Coor_paterno'];
        $_SESSION['MaternoCoord']=$row['Coor_materno'];
        header('Location:Captural.php');
    }
}
else {
    $_SESSION['usuario'] = $_POST['usuario'];
    $_SESSION['contrasena'] = $_POST['contrasena'];
    header('Location:index.php');
}

?>
```

Fig. 5.10 Autenticación con password, usuario y captcha.

5.3.2 Protección de la BD del Sistema

La BD del Sistema contiene la información de la representación de un partido político, sería crucial si esta información "apareciera" en otras manos, para cumplir con el requerimiento referente a la seguridad del sistema, implementaremos 4 mecanismos de seguridad aplicados a la BD del Sistema y al SGBD.

5.3.2.1 Cambiar el root de MySQL

Cuando se instala MySQL por primera ocasión es necesario configurar la base de datos llamada mysql, en especial la tabla que contiene los usuarios y los permisos o privilegios sobre las BD de MySQL. Por default esta tabla de usuarios trae un súper usuario llamado root, que en el inicio no tiene contraseña y tiene todos los privilegios sobre todas las bases de datos de MySQL, por lo que cualquier persona se podría conectar a el SGBD como el root sin ninguna contraseña y recibirá todos los privilegios. Además la instalación

del SGBD MySQL en ambiente Windows, genera otros 2 súper usuarios anónimos con los mismos privilegios que el root. Dados estos dos súper usuarios que se crean en nuestro gestor de base de datos, para proteger al mismo, al root le asignaremos una contraseña y desactivaremos la cuenta del súper usuario anónimo (figuras 5.11 a 5.13).

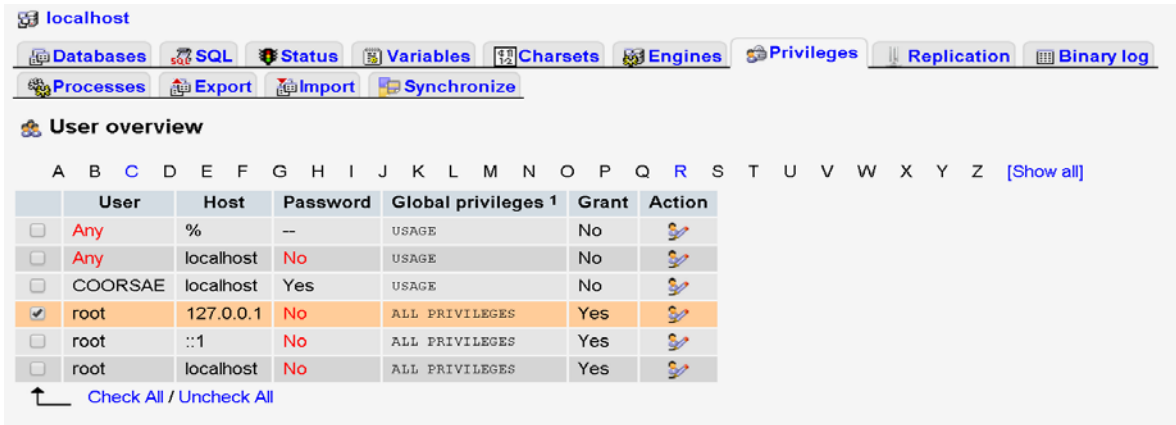


Fig. 5.11 Usuarios de MySQL

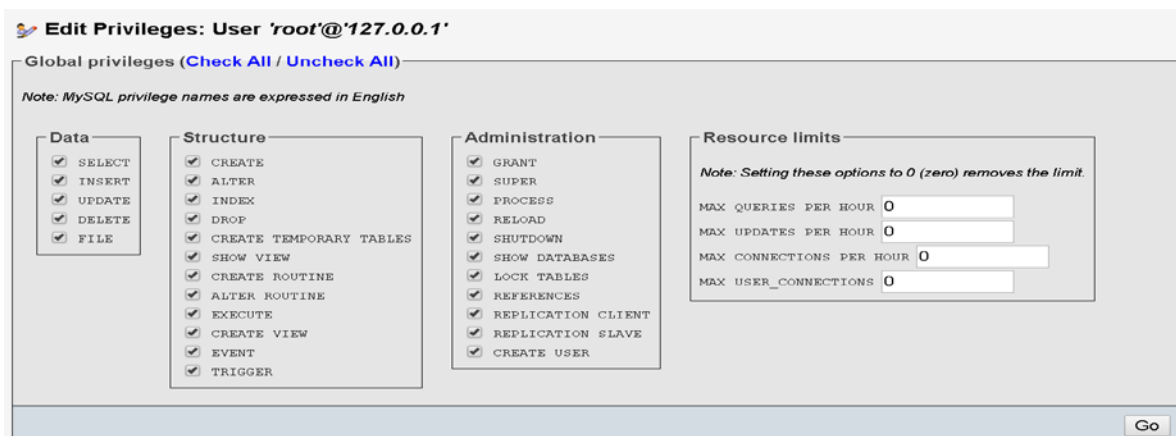


Fig. 5.12 Privilegios del root

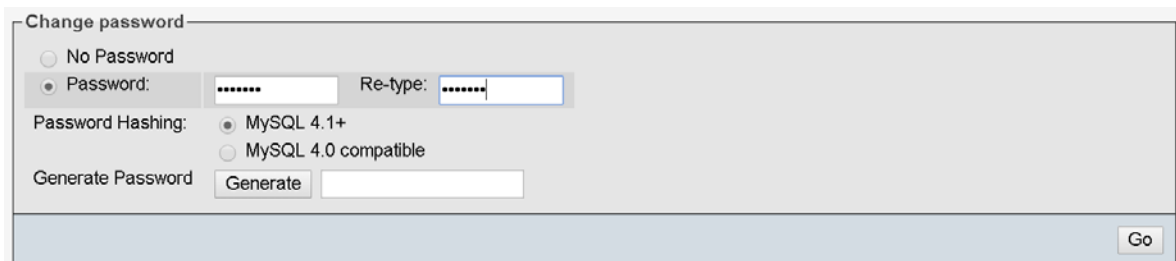


Fig. 5.13 Privilegios del root

5.3.2.2 Restricción de privilegios a los usuarios

El administrador del sistema y el COORSAE son los dos tipos de actores que tenemos definidos, para evitar intrusiones a la BD a través de inyección de código, deberemos restringir los privilegios a estos actores dejando los mínimos para el buen funcionamiento del sistema.

En la tabla 5.1 describimos los privilegios que tiene cada actor con respecto de las tablas de la Base de Datos

TABLA	PRIVILEGIOS	
	COORSAE	ADMINISTRADOR
1 LISTANOMINAL	SELECT	SELECT, INSERT, UPDATE
2 REPRESENTANTES	NINGUNO	SELECT, INSERT, UPDATE
6 COORDINADOR	NINGUNO	SELECT, INSERT, UPDATE
7 PROM_VOTO	SELECT	SELECT, INSERT, UPDATE
8 INSACULADO	SELECT	SELECT, INSERT, UPDATE
9 OTROSP	SELECT	SELECT, INSERT, UPDATE
10 DISTRITO	SELECT	SELECT, INSERT, UPDATE
11 MUNICIPIO	SELECT	SELECT, INSERT, UPDATE
12 SECCION	SELECT	SELECT, INSERT, UPDATE
13 CASILLA	SELECT	SELECT, INSERT, UPDATE
14 TELÉFONOS	NINGUNO	SELECT, INSERT, UPDATE

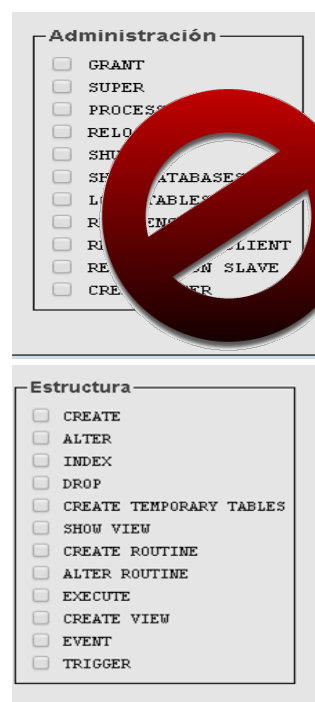


Tabla 5.1 Privilegios para los actores sobre las tablas de la BD

En la figura 5.15 mostramos cómo se crea el usuario que representa al actor "administrador" y cómo se le asignan los privilegios previstos en la tabla 5.14. Solo presentamos un ejemplo de la asignación de privilegios sobre la tabla casillas, este es presentado en la figura 5.16, este proceso se repitió para la restricción de los privilegios del actor COORSAE.

Add a new User

Login Information

User name: Use text field: AdmonSAE

Host: Local localhost¹

Password: Use text field:

Re-type:

Generate Password: Generate

Database for user

None
 Create database with same name and grant all privileges
 Grant all privileges on wildcard name (username_%)

Global privileges (Check All / Uncheck All)

Note: MySQL privilege names are expressed in English

Data <input checked="" type="checkbox"/> SELECT <input checked="" type="checkbox"/> INSERT <input checked="" type="checkbox"/> UPDATE <input checked="" type="checkbox"/> DELETE <input checked="" type="checkbox"/> FILE	Structure <input type="checkbox"/> CREATE <input type="checkbox"/> ALTER <input type="checkbox"/> INDEX <input type="checkbox"/> DROP <input type="checkbox"/> CREATE TEMPORARY TABLES <input type="checkbox"/> SHOW VIEW <input type="checkbox"/> CREATE ROUTINE <input type="checkbox"/> ALTER ROUTINE <input type="checkbox"/> EXECUTE <input type="checkbox"/> CREATE VIEW <input type="checkbox"/> EVENT <input type="checkbox"/> TRIGGER	Administration <input type="checkbox"/> GRANT <input type="checkbox"/> SUPER <input type="checkbox"/> PROCESS <input type="checkbox"/> RELOAD <input type="checkbox"/> SHUTDOWN <input type="checkbox"/> SHOW DATABASES <input type="checkbox"/> LOCK TABLES <input type="checkbox"/> REFERENCES <input type="checkbox"/> REPLICATION CLIENT <input type="checkbox"/> REPLICATION SLAVE <input type="checkbox"/> CREATE USER	Resource limits Note: Setting these options to 0 (zero) removes the limit. MAX QUERIES PER HOUR: 0 MAX UPDATES PER HOUR: 0 MAX CONNECTIONS PER HOUR: 0 MAX USER_CONNECTIONS: 0
---	---	--	--

Database-specific privileges

Database	Privileges	Grant	Table-specific privileges	Action
None				

Add privileges on the following database: Use text field: sae¹

Fig. 5.15 Asignación de privilegios al administrador desde phpmyAdmin, sobre la base de datos

localhost

Databases SQL Status Variables Charsets Engines Privileges Replication

Edit Privileges: User 'AdmonSAE'@'localhost' - Database sae - Table casilla

Table-specific privileges¹

SELECT	INSERT	UPDATE	REFERENCES	DELETE
Seccion	Seccion	Seccion	Seccion	<input checked="" type="checkbox"/>
Tipo	Tipo	Tipo	Tipo	<input checked="" type="checkbox"/>
RC_sup	RC_sup	RC_sup	RC_sup	<input checked="" type="checkbox"/>
RC_prop	RC_prop	RC_prop	RC_prop	<input checked="" type="checkbox"/>
RG_prop	RG_prop	RG_prop	RG_prop	<input checked="" type="checkbox"/>
RG_sup	RG_sup	RG_sup	RG_sup	<input checked="" type="checkbox"/>
Or <input type="checkbox"/> None	Or <input type="checkbox"/> None	Or <input type="checkbox"/> None	Or <input type="checkbox"/> None	<input checked="" type="checkbox"/> CREATE
				<input checked="" type="checkbox"/> DROP
				<input checked="" type="checkbox"/> GRANT
				<input checked="" type="checkbox"/> INDEX
				<input checked="" type="checkbox"/> ALTER
				<input checked="" type="checkbox"/> CREATE_VIEW
				<input checked="" type="checkbox"/> SHOW_VIEW
				<input checked="" type="checkbox"/> TRIGGER

[Database sae: Structure] [Table casilla: Browse]

¹ Note: MySQL privilege names are expressed in English

Fig. 5.16 Ejemplo de asignación de privilegios al administrador desde phpmyAdmin, sobre la tabla casilla

5.3.2.3 Procedimientos almacenados

Un procedimiento almacenado es un conjunto de comandos SQL que son almacenados en el servidor de BD, una vez que ya hemos programado y guardado un procedimiento podemos solo referirnos a él, sin necesidad de que en el cuerpo del programa estemos relanzando las sentencias SQL por separado. Para hacer las consultas requeridas en el desarrollo del sistema, desde PHP utilizaremos el llamado a procedimientos, por lo que desde el SGBD MySQL crearemos los siguientes procedimientos para hacer consultas a la BD. Esta técnica nos garantizará que las aplicaciones y los usuarios no tendrán ningún acceso directo a las tablas de la base de datos, sólo podrán ejecutar los procedimientos almacenados necesarios. En la siguiente figura presentamos un procedimiento almacenado para el módulo de insertar representante de casilla, este solo es un pequeño ejemplo del manejo de procedimientos almacenados que se usaron el sistema (figura 5.17).

```
BDyProcAlmacenados: Bloc de notas
Archivo Edición Formato Ver Ayuda
-- PROCEDIMIENTO PARA GUARDAR REGISTROS
USE sae ;
DROP PROCEDURE IF EXISTS guarda_registro;
DELIMITER $$
CREATE PROCEDURE guarda_registro(
    IN pclaveelector VARCHAR(18),
    IN pcargo VARCHAR(20),
    IN pprop_sup VARCHAR(20),
    IN pseccion VARCHAR(20),
    IN pcasilla VARCHAR(20),
    IN pnombre VARCHAR(50),
    IN papepaterno VARCHAR(50),
    IN papematerno VARCHAR(50),
    IN pcalle VARCHAR(80),
    IN pinterior VARCHAR(20),
    IN pexterior VARCHAR(20),
    IN pcolonia VARCHAR(80),
    IN pmunicipio VARCHAR(40)
)
BEGIN
    CASE pcargo
        WHEN 'RC' THEN
            insert into REPRESENTANTES(claveelector,Nombre,ApePaterno,ApeMaterno,Calle,NoInterior,
                NoExterior,Colonia,Municipio,Prop_Sup)
            values(pclaveelector,pnombre,papepaterno,papematerno,pcalle,pinterior,
                pexterior,pcolonia,pmunicipio,pprop_sup);
            if pprop_sup='Prop' then
                update casilla set casilla.RC_prop=pclaveelector where seccion=pseccion and tipo=pcasilla;
            else
                update casilla set casilla.RC_sup=pclaveelector where seccion=pseccion and tipo=pcasilla;
            end if;
        WHEN 'RG' THEN
            insert into rg(claveelector,Nombre,ApePaterno,ApeMaterno,Calle,NoInterior,
                NoExterior,Colonia,Municipio,Prop_Sup)
            values(pclaveelector,pnombre,papepaterno,papematerno,pcalle,pinterior,
                pexterior,pcolonia,pmunicipio,pprop_sup);
            if pprop_sup='Prop' then
                update casilla set casilla.RG_prop=pclaveelector where seccion=pseccion and tipo=pcasilla;
            else
                update casilla set casilla.RG_sup=pclaveelector where seccion=pseccion and tipo=pcasilla;
            end if;
    END CASE;
END $$
```

Fig. 5.17 Script de procedimiento almacenado para insertar a la BD

5.3.2.4 Encriptado de información

Otra técnica de seguridad en BD que empleamos en el sistema, fue usar un algoritmo asimétrico de cifrado de información por bloques llamado AES, que implementamos con PHP (figura 5.18). La información se guarda cifrada en la BD y cuando sea necesario hacer una consulta se descifran los datos para ser mostrados al usuario. En las siguientes figuras mostramos los scripts de la implementación del AES.

Mostrando registros 0 - 2 (+3) total, La consulta tardó 0.0003 seg

Mostrar: 30 filas empezando de 0 en modo horizontal y repetir los encabezados cada 100 celdas

Organizar según la clave: Ninguna

	claveelector	Nombre	ApePaterno	ApeMaterno	Calle	NoInterior	NoExterior	Colonia	Municipio	Tipo	CasillaSeccion	CasillaTipo
<input type="checkbox"/>	EEF-0ÁmÉ7ó'æp#4#	EAQ,UM	ED,ROEÍB	ED,ROEÍB	ES°_7K**U)Aááwó #æ±		Et ÚDáVn	EQÜX(é> ZHh&SóM	EQÜX(é> ZHh&SóMÜ-t	€1BWDæJ&	€:áÀ°	Ed-g-¼
<input type="checkbox"/>	€Í\$0TUé>óÆáBÓ óáY	EAQ,UM	€4Ér4iy6	€4Ér4iy6	€c nYNSÁN@	€-UEq-(UæU é°Iz>	€c'05...Éo	€óXU%2_0µeQT	€RE%b\$ç>10BY.ÓV° (F	€'MSYDk	€ htæy0'	€K 4FAV
<input type="checkbox"/>	€'Ó**x)kT30UáU'i	€ÁæÁ&ç	€SZVLL±'U	€ip JÇ	€ANUQá.Íló.éÁ'@-		€sç'5íE	€i.áV(6BO)jÓ†}_5	€b%4n ~\$#máæURU)	€'MSYDk	€50æKç0Ú	€uü0NÚ

Fig. 5.18 Encriptado de información en el momento de insertar sobre la tabla REPRESENTANTE

Hasta aquí damos por concluido el 1^{er} sprint, cumpliendo con los requisitos del Backlog. En las siguientes secciones solo citaremos el Backlog de cada sprint sin explicar a fondo la implementación de cada requerimiento.

5.4 2^{do} Sprint

- ✓ Elaborar los programas que hagan posible el realizar cambios en una fecha límite del requerimiento RF3.1.
- ✓ Modificar el módulo para realizar eliminaciones múltiples en la tabla REPRESENTANTES.
- ✓ Elaborar el módulo que permita generar el reporte impreso del avance de estructura electoral, para dar cumplimiento al requerimiento RF7.1
- ✓ Permitir que este reporte se genere dependiendo del actor y que muestre con una gráfica de pastel el avance de estructura.

5.5 3^{er} Sprint

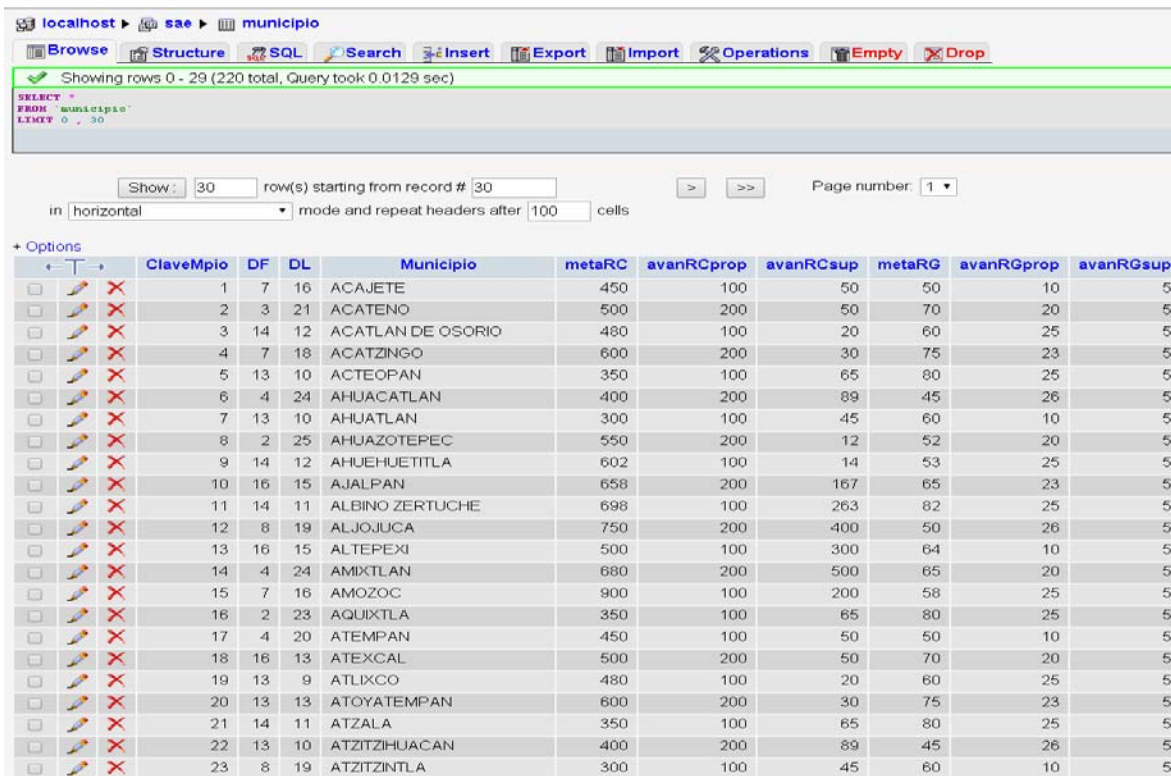
- ✓ Elaborar el módulo que genere la impresión de la solicitud de registro del representante, este módulo deberá ser de impresión múltiple, es decir que el actor pueda seleccionar de la lista de representantes dados de alta en el sistema, a quienes se les deberá imprimir la solicitud.

CAPÍTULO 6 PRUEBAS

En esta sección, después de haber avanzado con 3 sprints, probaremos y validaremos al sistema con los requisitos especificados en el Backlog del producto, para lo que necesariamente convertiremos y cargaremos la información requerida a la BD.

6.1 CONVERSIÓN Y CARGA DE DATOS

Se dice que la etapa de conversión y carga de datos solo se realiza cuando se reemplaza un sistema antiguo por uno nuevo, en este caso, aunque no existe un sistema de software antiguo que se encargara de la administración de los representantes, pues se usaban hojas de cálculo para el almacén de la información, fue necesario convertir y cargar algunas de las tablas a la base de datos de Sistema Electoral. En la figura 6.1 mostramos ejemplo de la carga de datos de la tabla de municipios, esta información corresponde a los 217 municipios del estado de Puebla.



The screenshot shows a database management interface with a table of municipalities. The table has the following columns: ClaveMpio, DF, DL, Municipio, metaRC, avanRCprop, avanRCsup, metaRG, avanRGprop, and avanRGsup. The data is displayed in a grid format with 23 rows visible. Each row includes a checkbox, a pencil icon, and a red 'X' icon. The 'Municipio' column lists various municipalities such as ACAJETE, ACATENO, ACATLAN DE OSORIO, ACATZINGO, ACTEOPAN, AHUACATLAN, AHUATLAN, AHUAZOTEPEC, AHUEHUETITLA, AJALPAN, ALBINO ZERTUCHE, ALJOJUCA, ALTEPEXI, AMIXTLAN, AMOZOC, AQUIXTLA, ATEMPAN, ATEXCAL, ATLIXCO, ATOYATEMPAN, ATZALA, ATZITZIHUACAN, and ATZITZINTLA.

	ClaveMpio	DF	DL	Municipio	metaRC	avanRCprop	avanRCsup	metaRG	avanRGprop	avanRGsup
<input type="checkbox"/>	1	7	16	ACAJETE	450	100	50	50	10	5
<input type="checkbox"/>	2	3	21	ACATENO	500	200	50	70	20	5
<input type="checkbox"/>	3	14	12	ACATLAN DE OSORIO	480	100	20	60	25	5
<input type="checkbox"/>	4	7	18	ACATZINGO	600	200	30	75	23	5
<input type="checkbox"/>	5	13	10	ACTEOPAN	350	100	65	80	25	5
<input type="checkbox"/>	6	4	24	AHUACATLAN	400	200	89	45	26	5
<input type="checkbox"/>	7	13	10	AHUATLAN	300	100	45	60	10	5
<input type="checkbox"/>	8	2	25	AHUAZOTEPEC	550	200	12	52	20	5
<input type="checkbox"/>	9	14	12	AHUEHUETITLA	602	100	14	53	25	5
<input type="checkbox"/>	10	16	15	AJALPAN	658	200	167	65	23	5
<input type="checkbox"/>	11	14	11	ALBINO ZERTUCHE	698	100	263	82	25	5
<input type="checkbox"/>	12	8	19	ALJOJUCA	750	200	400	50	26	5
<input type="checkbox"/>	13	16	15	ALTEPEXI	500	100	300	64	10	5
<input type="checkbox"/>	14	4	24	AMIXTLAN	680	200	500	65	20	5
<input type="checkbox"/>	15	7	16	AMOZOC	900	100	200	58	25	5
<input type="checkbox"/>	16	2	23	AQUIXTLA	350	100	65	80	25	5
<input type="checkbox"/>	17	4	20	ATEMPAN	450	100	50	50	10	5
<input type="checkbox"/>	18	16	13	ATEXCAL	500	200	50	70	20	5
<input type="checkbox"/>	19	13	9	ATLIXCO	480	100	20	60	25	5
<input type="checkbox"/>	20	13	13	ATOYATEMPAN	600	200	30	75	23	5
<input type="checkbox"/>	21	14	11	ATZALA	350	100	65	80	25	5
<input type="checkbox"/>	22	13	10	ATZITZIHUACAN	400	200	89	45	26	5
<input type="checkbox"/>	23	8	19	ATZITZINTLA	300	100	45	60	10	5

Fig. 6.1 Carga de datos de la tabla municipios

En la figura 6.2 se muestra la carga del catálogo de las 6 mil casillas

Showing rows 0 - 29 (6,912 total, Query took 0.0004 sec)

```

SELECT *
FROM `casilla`
LIMIT 0, 30

```

Show: 30 row(s) starting from record # 30 in horizontal mode and repeat headers after 100 cells

+ Options

			Seccion	Tipo	RC_sup	RC_prop	RG_prop	RG_sup
<input type="checkbox"/>			535	B				
<input type="checkbox"/>			535	C1				
<input type="checkbox"/>			535	C2				
<input type="checkbox"/>			536	B				
<input type="checkbox"/>			536	C1				
<input type="checkbox"/>			536	C2				
<input type="checkbox"/>			537	B				
<input type="checkbox"/>			537	C1				
<input type="checkbox"/>			537	EX1				
<input type="checkbox"/>			538	B				
<input type="checkbox"/>			538	EX1				
<input type="checkbox"/>			538	EX2				
<input type="checkbox"/>			539	B				
<input type="checkbox"/>			539	C1				
<input type="checkbox"/>			540	B				
<input type="checkbox"/>			540	EX1				
<input type="checkbox"/>			541	B				
<input type="checkbox"/>			541	C1				
<input type="checkbox"/>			541	C2				
<input type="checkbox"/>			542	B				
<input type="checkbox"/>			542	C1				
<input type="checkbox"/>			543	B				
<input type="checkbox"/>			502	B				
<input type="checkbox"/>			502	C1				

Fig. 6.2 Carga de datos de la tabla casillas

La tabla que requirió mayor atención para la conversión y carga fue la de lista de nombres (LN), pues la información viene contenida en un archivo de texto y contiene miles de registros. MySQL no tiene alguna herramienta que permita la carga automática de tablas de estas dimensiones, una solución que encontramos fue crear un programa que leyera el archivo de texto y lo convirtiera en un script de SQL y posteriormente ejecutamos este script desde la consola de MySQL. Una vez terminada la conversión y carga de datos podremos comenzar con las pruebas al sistema.

6.2 PRUEBAS

En esta sección realizamos pruebas al funcionamiento del Sistema con la carga de datos que describimos en la sección anterior, estas se aplicaron a cada módulo del sistema, como a continuación describimos.

6.2.1 Captura de representante

En este módulo se lleva a cabo el ingreso de información general de los representantes, así como la asignación de cargo, tipo y en caso de ser representante de casilla (RC) la asignación de la casilla o casillas si es representante general (RG) y distrito en caso de ser representante ante consejo distrital (RCDE) o de comisión de vigilancia (RCVE) (figura 6.3).

La captura se inicia con la inserción de la clave de elector en el formulario siguiente.

The screenshot shows a web browser window with the URL `localhost/SAE1.2/Captura1.php`. The page header features the SAE logo and the title "Sistema de Administración de la Estructura" with the slogan "Por una representación comprometida". Below the header is a navigation menu with buttons for "CAPTURAR", "ACTUALIZAR", "BUSCAR", "ESTADÍSTICAS", and "SALIR". The main content area displays a welcome message: "BIENVENIDO COORDINADOR: Mateo Pakiao Mata". On the left, there is a sidebar menu with "Ver Representantes" and sub-links for "generales", "De Casilla", and "De Casilla por Sección". The main form is titled "DATOS GENERALES" and contains the following fields:

- Clave de elector:
- Nombre:
- Apellido Paterno:
- Apellido Materno:

Below this is the "DOMICILIO" section with the following fields:

- Calle:
- Número: Ext: Int:
- Colonia:
- Municipio:

Fig. 6.3 Ejemplo de captura de representante

Para facilitar este proceso se realiza la búsqueda de la clave de elector en la tabla que contiene la lista de nombres (LN) y el sistema llena el formulario con la información correspondiente, como se muestra en la fig. 6.4.

localhost/SAE1.2/Captura2.php

SAE "Por una representación comprometida"

ENTRAR CAPTURAR ACTUALIZAR BUSCAR ESTADÍSTICAS SALIR

BIENVENIDO COORDINADOR: Mateo Pakiao Mata

Ver Representantes

[Generales](#)
[De Castilla](#)
[De Castilla por Sección](#)

DATOS GENERALES

Clave de elector: BABAIS78070621H500

Nombre: ISAIAS

Apellido Paterno: MAYO

Apellido Materno: MAYO

DOMICILIO

Calle: C AMADO

Número: Ext: 700 Int:

Colonia: CENTRO

Municipio: PUE.

ASIGNACIÓN DE CARGO

Cargo: RC Tipo: Propietario

Seleccionar cargo

Casilla:

Fig. 6.4 Ejemplo de captura de representante

El siguiente paso es la asignación de cargo, en este ejemplo vamos a establecer el cargo RC propietario y le asignamos la casilla 3 C1, es importante mencionar que el combo que muestra las casillas, solo muestra las casillas correspondientes disponibles para este cargo, reduciendo con esto la posibilidad de que existan 2 o más representantes repetidos con el mismo cargo en la misma casilla, esto lo mostramos en la fig. 6.5.

The screenshot shows a web browser window with the URL localhost/SAE1.2/Captura3.php. The application header includes the SAE logo and the slogan "Por una representación comprometida". A navigation bar contains buttons for ENTRAR, CAPTURAR, ACTUALIZAR, BUSCAR, ESTADÍSTICAS, and SALIR. The main content area displays a welcome message for the coordinator, Mateo Pakiao Mata, and a form for entering representative data.

Ver

DATOS GENERALES

Clave de elector: BABAIS78070621H500

Nombre: ISAIAS

Apellido Paterno: MAYO

Apellido Materno: MAYO

DOMICILIO

Calle: C AMADO

Número: Ext: 700 Int:

Colonia: CENTRO

Municipio: PUE.

ASIGNACIÓN DE

Cargo: RC

Tipo:

Casilla: 3 C1

Guardar Representante

The dropdown menu for 'Casilla' is open, showing a list of options: 3 B, 3 C1 (selected), 3 C2, 4 B, 4 C1, 4 C2, 5 B, 5 C1, 5 C2, 5 C3, 6 B, 6 C1, 6 C2, 7 B, 7 C1, 7 C2, 7 C3, 8 B, 8 C1, 8 C2.

Fig. 6.5 Ejemplo de captura de representante

Finalmente se guarda la información del representante, en caso de que este representante ya estuviera dado de alta como promotor del voto o representante de otro partido e inclusive del mismo, el sistema manda un mensaje alertando que el representante ya está dado de alta en otra estructura. En la figura 6.6 mostramos para el caso de que este representante ya esté dado de alta como representante del mismo partido.

Fig. 6.6 Ejemplo de captura de representante

6.2.3 Eliminación de representante

Este módulo permite hacer desde una hasta múltiples eliminaciones y solo permite eliminar de la tabla de representantes. Para dar de baja un representante primero, se elimina de la tabla de representantes, segundo en la tabla casilla se borra la clave de elector de la casilla o casillas en las que había sido asignado.

BIENVENIDO COORDINADOR: DEYANIRA PEREZ GONZALEZ DISTRITO 7

Borrar

RG

RC

Seleccionar para borrar

BORRAR	PROG	CARGO	TIPO	SECCION	CASILLA	NOMBRE
<input type="checkbox"/>	1	RC	Prop	1	B	ADELA ABADIA ABADIA
<input type="checkbox"/>	2	RC	Prop	1	C	ADELA ABURTO ABURTO
<input type="checkbox"/>	3	RC	Prop	2	C1	ANTONIO ABRAHAM ABITI
<input type="checkbox"/>	4	RC	Prop	2	C2	ARIEL ABRAHAM ABURTO
<input type="checkbox"/>	5	RC	Prop	2	B	ARTEMA ABRIL ABRIL
<input type="checkbox"/>	6	RC	Prop	3	B	MARIA CHARLOT ABITI Y ABRAHAM
<input type="checkbox"/>	7	RC	Prop	3	C1	CLAUDIA EVELDA ABELAR ABELAR
<input type="checkbox"/>	8	RC	Prop	3	C2	CINDY SELENE ABDALA ABRAHAM

BORRAR

Fig. 6.7 Ejemplo de eliminación de representante

6.2.4 Consulta de representante

Este módulo se descompone en tres opciones para mostrar la información contenida en la tabla de representante: 1) Muestra listado de RG (fig. 6.8), 2) Muestra listado de RC (fig. 6.9) y 3) muestra RC por sección electoral (fig. 6.10).

The screenshot shows the SAE system interface. At the top, there is a navigation bar with buttons for CAPTURAR, ACTUALIZAR, BUSCAR, ESTADÍSTICAS, and SALIR. Below this, a welcome message reads 'BIENVENIDO COORDINADOR: Mateo Pakiao Mata'. On the left, there is a sidebar with 'Ver Representantes' and sub-options: 'Generales', 'De Casilla', and 'De Casilla por Sección'. The main content area displays the 'LISTADO DE REPRESENTANTES GENERALES' table.

PROG	SECCION	CASILLA	NOMBRE	TIPO
1	1	B	MINDY SELENE ABDALA ABRAJAN	Prop
2	1	B	CARMEN ABAD ABURTO	Sup
3	1	C1	FRIACCO ABURTO	Sup
4	2	B	ISAIAS ABRIL ABRIL	Sup

Fig. 6.8 Muestra listado de RG

The screenshot shows the SAE system interface for the 'LISTADO DE REPRESENTANTES DE CASILLA'. It features the same navigation bar and welcome message as Fig. 6.8. The sidebar shows 'Ver Representantes' with sub-options: 'Generales', 'De Casilla', and 'De Casilla por Sección'. The main content area displays the 'LISTADO DE REPRESENTANTES DE CASILLA' table.

PROG	SECCION	CASILLA	NOMBRE	TIPO
1	1	B	AUREA ABAHA ASADIA	Prop
2	1	B	ACELA ABURTO ABURTO	Sup
3	1	C1	ANTONIO ABRAHAM ABITI	Prop
4	1	C1	ARIEL ABAD ABURTO	Sup
5	1	C2	ARTEMIA ABRIL ABRIL	Prop
6	1	C2	MARIA CHARLOT ABITI	Sup
7	2	B	CLAUDIA MELIDA ABELAR ABELAR	Prop
8	2	B	DORA MARIA ABURTO	Sup
9	2	C1	FERNANDO ABURTO	Prop
10	2	C2	MINGO ABURTO ABURTO	Prop

Fig. 6.9 Muestra listado de RC



Fig. 6.10 Muestra listado de RC por sección electoral



Fig. 6.10 Muestra listado de RC por sección electoral

6.2.5 Estadísticas avance de estructura por distrito

A través de gráficas se muestra el avance de estructura de representantes generales y de casilla como se muestra en la fig. 6.11.



Fig. 6.11 Ejemplo de consulta de avance de estructura de RG del distrito 1 Huauchinango.

El sistema también ofrece la opción de generar este mismo reporte en PDF y así proporciona la opción de imprimir este reporte, esto lo mostramos en las siguientes figuras.

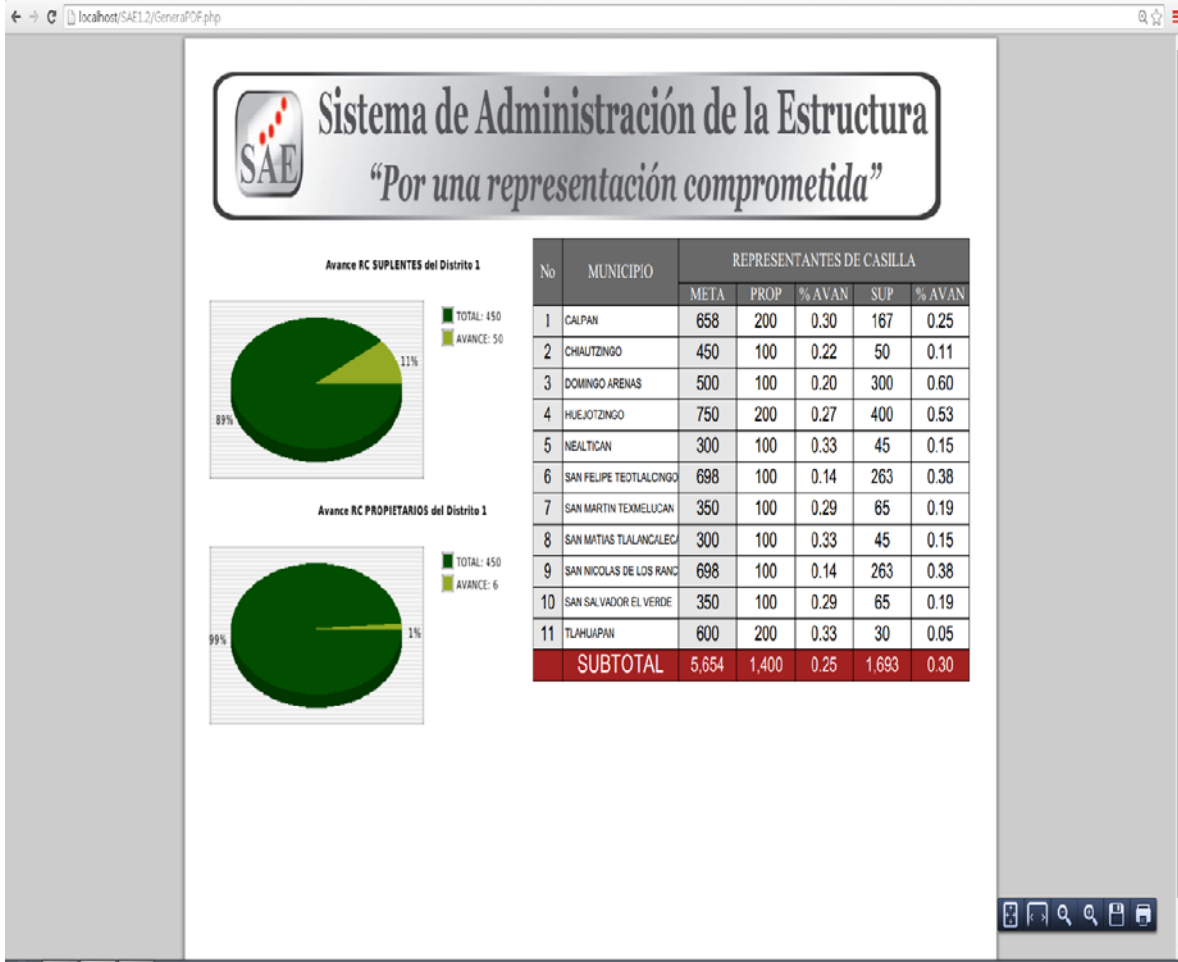


Fig. 6.12 Ejemplo de reporte de avance de estructura RG en PDF del distrito 1 de Huauchinango.

Cabe recordar que los reportes se muestran dependiendo del actor que está usando el sistema, pues se generan sesiones diferentes para cada actor, así pues, si el COORSAE del distrito 7 de Tepeaca ingresa al sistema se mostrarán los reportes solo del distrito 7.

6.2.5 Estadísticas avance de estructura Estatal

El acceso a este módulo solo es permitido para el Administrador, pues muestra un reporte de avance por cada distrito en PDF, desglosando los municipios que pertenecen a cada distrito, permite la impresión y guardado de archivo en caso de querer llevar un historial de avance.

Sistema de Administración de la Estructura
“Por una representación comprometida”

AVANCE DE ESTRUCTURA REPORTE DISTRITAL

DF	No	MUNICIPIO	REPRESENTANTES DE CASILLA				REPRESENTANTES GENERALES					
			META	PROP	% AVAN	SUP	% AVAN	META	PROP	% AVAN	SUP	% AVAN
1	1	CHICHONJAUHTA	500	200	0.40	50	0.10	70	20	0.29	5	0.07
1	2	FRANCISCO Z. MENA	450	100	0.22	50	0.11	50	10	0.20	5	0.10
1	3	HONEY	658	200	0.30	167	0.25	65	23	0.35	5	0.08
1	4	ILUUCIINANGO	550	200	0.36	12	0.02	52	20	0.38	5	0.10
1	5	JALPAN	550	200	0.36	12	0.02	52	20	0.38	5	0.10
1	6	JOPALA	696	100	0.14	263	0.38	82	25	0.30	5	0.06
1	7	JUAN GALINDO	500	100	0.20	300	0.60	64	10	0.16	5	0.08
1	8	NAUPAN	350	100	0.29	65	0.19	80	25	0.31	5	0.06
1	9	PAULATLAN	680	200	0.29	500	0.74	65	20	0.31	5	0.08
1	10	PANTEPEC	350	100	0.29	65	0.19	80	25	0.31	5	0.06
1	11	TLACUILOTEPEC	500	200	0.40	50	0.10	70	20	0.29	5	0.07
1	12	TLAOLA	300	100	0.33	45	0.15	60	10	0.17	5	0.08
1	13	TLAXCO	696	100	0.14	263	0.38	82	25	0.30	5	0.06
1	14	VIPALUSTIANO CARRANZA	500	200	0.40	50	0.10	70	20	0.29	5	0.07
1	15	XICOTEPEC DE JUAREZ	350	100	0.29	65	0.19	80	25	0.31	5	0.06
1	16	ZHAIATZUTLA	350	100	0.29	65	0.19	80	25	0.31	5	0.06
SUBTOTAL			7,984	2,300	0.29	2,022	0.25	1,102	323	0.29	80	0.07

6 Octubre 2014, 9:38 pm Page 2/20

Fig. 6.13 Ejemplo de reporte de avance de estructura por distrito

En este módulo también se muestra un reporte que contiene el concentrado por distrito electoral de todo el estado.

Sistema de Administración de la Estructura
"Por una representación comprometida"

AVANCE DE ESTRUCTURA REPORTE ESTATAL

DF	CABECERA	REPRESENTANTES DE CASILLA					REPRESENTANTES GENERALES				
		META	PRCP	% AVAN	SUP	% AVAN	META	PRCP	% AVAN	SUP	% AVAN
1	HUIACHINANGO	450	6	0.01	50	0.11	50	10	0.20	5	0.10
2	ZACATLAN	500	200	0.40	50	0.10	70	20	0.29	5	0.07
3	TEZUTLAN	480	100	0.21	20	0.04	60	25	0.42	5	0.08
4	ZACAPOAXTLA	600	200	0.33	30	0.05	75	23	0.31	5	0.07
5	SAN MARTIN TEXMELUCAN	350	100	0.29	65	0.19	80	25	0.31	5	0.06
6	PUEBLA	400	200	0.50	89	0.22	45	26	0.58	5	0.11
7	TEPEACA	300	100	0.33	45	0.15	60	10	0.17	5	0.08
8	CIUDAD SERDAN	550	200	0.36	12	0.02	52	20	0.38	5	0.10
9	PUEBLA	602	100	0.17	14	0.02	53	25	0.47	5	0.09
10	SAN PEDRO CHOLLULA	658	200	0.30	167	0.25	65	23	0.35	5	0.08
11	PUEBLA	698	100	0.14	263	0.38	82	25	0.30	5	0.06
12	PUEBLA	750	200	0.27	400	0.53	50	26	0.52	5	0.10
13	ATLIXCO	500	100	0.20	300	0.60	64	10	0.16	5	0.08
14	ZUCHAR DE MATAMOROS	680	200	0.29	500	0.74	65	20	0.31	5	0.08
15	TEHUACAN	900	100	0.11	200	0.22	58	25	0.43	5	0.09
16	AJALPAN	350	100	0.29	65	0.19	80	25	0.31	5	0.06
TOTAL		8,768	2,206	0.25	2,270	0.26	1,009	338	0.33	80	0.08

6 October 2014, 9:51 pm Page 1/1

Fig. 6.14 Ejemplo de reporte de avance de estructura por distrito

6.2.6 Generación de Oficio de Acreditación

En este apartado del sistema permite la visualización e impresión del oficio de acreditación para el representante, con esto queda concluido el requerimiento RF8.1

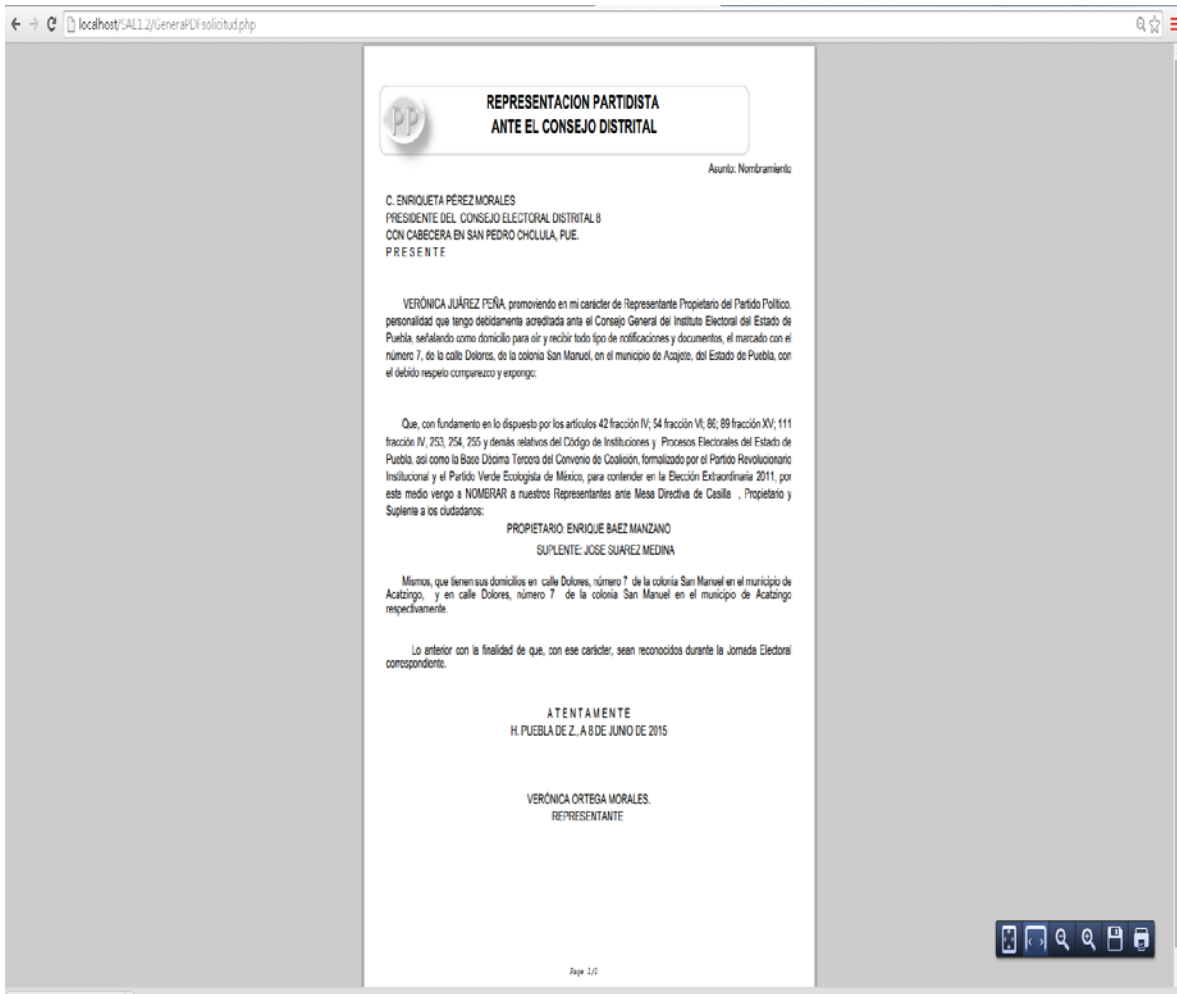


Fig. 6.15 Ejemplo de impresión de oficio para solicitud de registro de representante

6.3 Conclusión de las pruebas.

En esta fase verificamos a través de pruebas de caja negra, que externamente cumple con los requisitos del product backlog. Comprobamos que las funciones del software son operativas, que las entradas y salidas son correctas y que la integridad de la información se mantiene. En estas pruebas utilizamos los diagramas de secuencia que generamos en secciones anteriores y verificamos que las variantes de los casos de uso se reflejan en el sistema.

CONCLUSIONES

Elaborar un sistema de software sin el uso de alguna metodología nos haría difícil el camino a la meta. Para el desarrollo de este sistema utilizamos SCRUM, que aunque no es una metodología de desarrollo de software en sí, porque no define herramientas o lenguajes para la representación del análisis del sistema, es más bien una metodología orientada al desarrollo de proyectos, pero tiene varias ventajas sobre las metodologías clásicas de desarrollo de software. Una de estas ventajas, es que garantiza la entrega de un producto funcional al término de cada sprint. Otra es que origina que el grupo de trabajo esté comprometido con el proyecto, por lo que también garantiza la culminación de este.

Como SCRUM no define herramientas para las etapas principales del desarrollo de software, que son análisis, diseño, implementación y pruebas del sistema, para la descripción del análisis hicimos uso del lenguaje UML que nos permitió junto con los requerimientos del sistema tener un panorama general del mismo. De UML utilizamos los diagramas de contexto, de casos de uso, de clases y de secuencia, así como de los escenarios de casos de uso.

Para hacer el diseño del sistema a este lo descompusimos en 3 vertientes: 1)Diseño Arquitectónico, como se requería una arquitectura Cliente-Servidor, se utilizó Apache como servidor de páginas Web ; 2)Diseño de la Base de Datos, este dividido a su vez en tres etapas: diseño conceptual en el que utilizamos el Modelado EER, diseño lógico en el que se empleó el Modelo de datos Relacional y diseño físico en el que seleccionamos

nuestro SGBD MySQL; y 3)Diseño de la interfaz de usuario, en la que utilizamos HTML para la creación de las páginas web, CSS para el diseño de la interfaz y Adobe Illustrator para el diseño del logo y de las imágenes que se muestran en los menús.

La forma de trabajar de SCRUM es a través de iteraciones llamadas sprints, en las que al final de cada una de ellas se entrega un producto funcional. El 1er. sprint en el desarrollo de este software abarcó la parte del análisis, diseño y seguridad del sistema. En el 2do sprint se muestran avances en el módulo de captura de representantes y carga de la base de datos de electores. En el 3er. sprit reportes de captura y en el 4to. impresión del formato de solicitud de registro ante el IEE o el IFE. Con estos 4 sprints concluimos los requerimientos del Product Backlog, facilitando con este sistema la administración de información generada por los representantes de los partidos políticos. Pues con el módulo de seguimiento de la estructura, el Secretario puede estar al tanto de los avances en la conjunción de los representantes, así como del avance en la capacitación. Con el módulo de impresión de formato de solicitud de registro aligeramos la cantidad de trabajo de los coordinadores. Por lo que el proceso de registro de representantes ante los Institutos Electorales, desde la selección del representante hasta la impresión de su formato de solicitud será más sencillo.

TRABAJO FUTURO

Un trabajo futuro que se desprendería de este, sería agregar un módulo para llevar el conteo de votos para el día de la jornada electoral. Módulo que pueda detectar si en alguna casilla existen errores en el conteo de votos por parte de los integrantes de las mesas directivas de casillas, además de poder registrar si existieron incidentes en las casillas y notificar a la estructura de defensa del voto de alguna posible inconformidad.

Para el buen desempeño en época electoral los partidos políticos deben trabajar con estructuras, depende del partido político definir quién maneje qué estructura, o qué secretaría se haga cargo de una estructura definida. Por lo general los partidos políticos deben contar mínimo con dos tipos de estructuras la de promoción al voto y la de representación ante órganos electorales. Este sistema está enfocado a proporcionar servicios a la secretaría o entidad que se haga cargo de la representación oficial del partido, por lo que otro trabajo futuro sería adaptar un módulo para administrar servicios a la secretaría encargada de la promoción al voto.

Con el auge en el mercado de dispositivos móviles es necesario adaptar a los sistemas de software creados para PC's a estas nuevas tecnologías. El sistema que tenemos en desarrollo tiene proyectado adaptarse para poder ser ejecutado en dispositivos móviles.

BIBLIOGRAFÍA

- [1] Código Federal de Instituciones y Procedimientos Electorales, México, 2012.
- [2] Código de Instituciones y Procedimientos Electorales del Estado de Puebla, Puebla, 2013.
- [3] «SCRUM ORG,» 2012. [En línea]. Available: <https://www.scrum.org/>. [Último acceso: 27 Marzo 2012].
- [4] H. Kniberg, Scrum y XP desde las trincheras, EU: Projectalis, 2007.
- [5] A. S. Tanenbaum, Redes de Computadoras, Prentice Hall, 1997.
- [6] N. Elmasri, Fundamentos de Sistemas de Bases de Datos, Addison-Wesley, 2002.
- [7] A. De Miguel, M. Piattini y E. Marcos, Diseño de Bases de Datos Relacionales, Ra-Ma, 1999.
- [8] A. De Miguel y M. Piattini, Fundamentos y Modelos de Bases de Datos, Ra-Ma, 1999.
- [9] H. Saiedian, «Universidad de Chile,» 2002. [En línea]. Available: <http://users.dcc.uchile.cl/~cguierr/cursos/BD/extendido.pdf>. [Último acceso: 20 Julio 2014].
- [10] H. F. K. Abraham Silberschatz, FUNDAMENTOS DE BASES DE DATOS, España: Mc Graw Hill, 2002.
- [11] K. & Kendall, Análisis y diseño de Sistemas, México: Prentice Hall, 2011.

APÉNDICE DEL MARCO TEÓRICO

Clasificación	Comando	Uso
LDD	Create	Crea nuevas Bd's, tablas, campos e índices
	Drop	Elimina Bd's, tablas, e índices
	Alter	Modifica las tablas agregando campos
	Truncate	Trunca el contenido de una tabla
LMD	Insert	Inserta registros en una tabla
	Select	Muestra datos de una o varias tablas
	Update	Actualiza los registros de una tabla
	Delete	Borra uno o varios registros de una tabla
	Commint	Marca el final de una transacción correcta
	Rollback	Devuelve a la base de datos a algún estado previo
LCD	Grant	Permite a un usuarios ejecutar Transact-SQL
	Deny	Niega permisos sobre los objetos de la BD
	Revoke	Quita permisos concedidos.

Tabla A.1 Clasificación del lenguaje SQL

Cláusula	Uso
WHERE	Define la condición que deben cumplir los registros en el resultado de la consulta.
FROM	Lista las tablas que se usarán en la consulta.
INSERT INTO	Inserta un registro en una tabla.
JOIN	Vincula una o más tablas haciendo parecer que es una sola tabla.
ORDER BY	Ordena el resultado de alguna consulta, esta puede ser con base en una o más columnas.
GROUP BY	Agrupar todas las filas similares por algún campo definido y luego produce una fila de resultados por cada grupo.
HAVING	Clasifica solo aquellos grupos que cumplan alguna condición

Tabla A.2 Uso de las Cláusulas

Operador	Uso
<, >, <>, <=, >=, =	Compara mayor que, menor que, es diferente, menor o igual, mayor o igual e igual.
Between	Especifica un rango de valores y se usa en tipos de datos string, numéricos y de fechas.
Like	Se utiliza en la comparación de algún patrón. Cuenta con el carácter especial % que funciona como comodín y el comodín posicional _.
In	Verifica si el valor de algún campo se encuentra en una lista de valores.
Is	Verifica si algún atributo tiene valor nulo
Any	Compara cualquier valor de una lista
All	Compara todos los valores de una serie

Tabla A.3 Uso de los operadores de Comparación

Operador	Uso
AND	Concatena dos condiciones
OR	Vincula a dos condiciones con una disyuntiva
NOT	Negación de la siguiente expresión
XOR	Enlaza dos condiciones con una disyuntiva exclusiva

Tabla A.4 Uso de los operadores Lógicos

Función	Uso
AVD	Calcula el promedio
COUNT	Cuenta cantidad de registros
SUM	Suma los valores provenientes de una columna
MIN	Calcula el valor mínimo contenido en una columna
MAX	Calcula el valor máximo contenido en una columna
DATE	Muestra fecha

Tabla A.5 Uso de las Funciones Agregadas

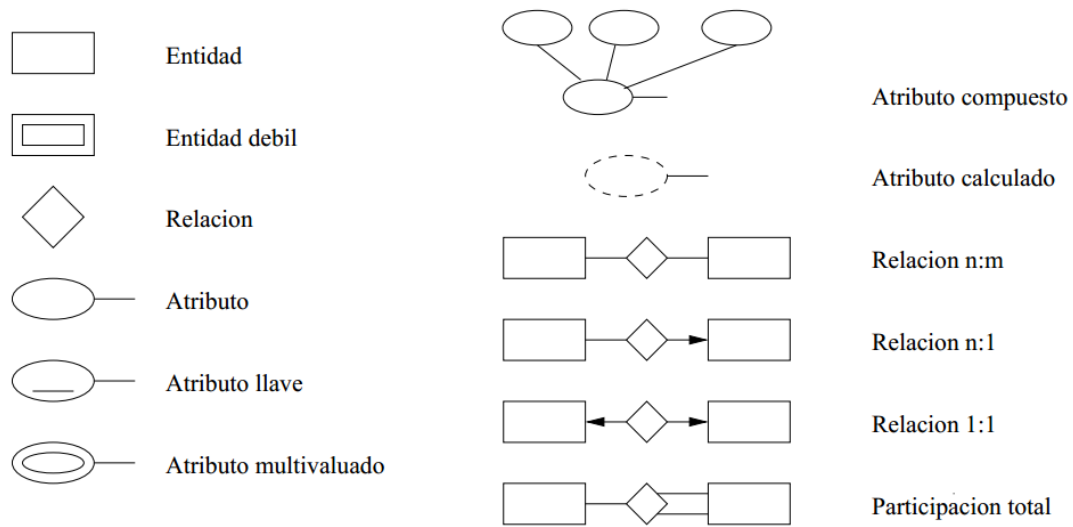


Figura A.6 Notación del Modelo Entidad Relación

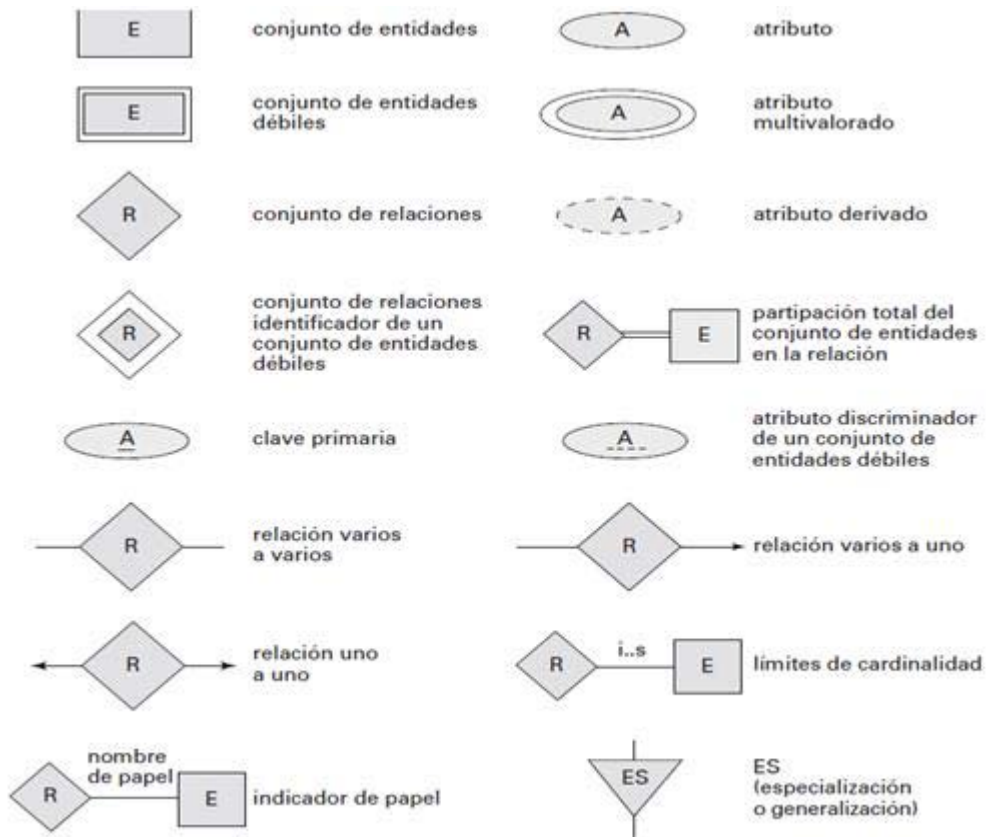


Figura A.7 Notación del Modelo Entidad Relación Extendido

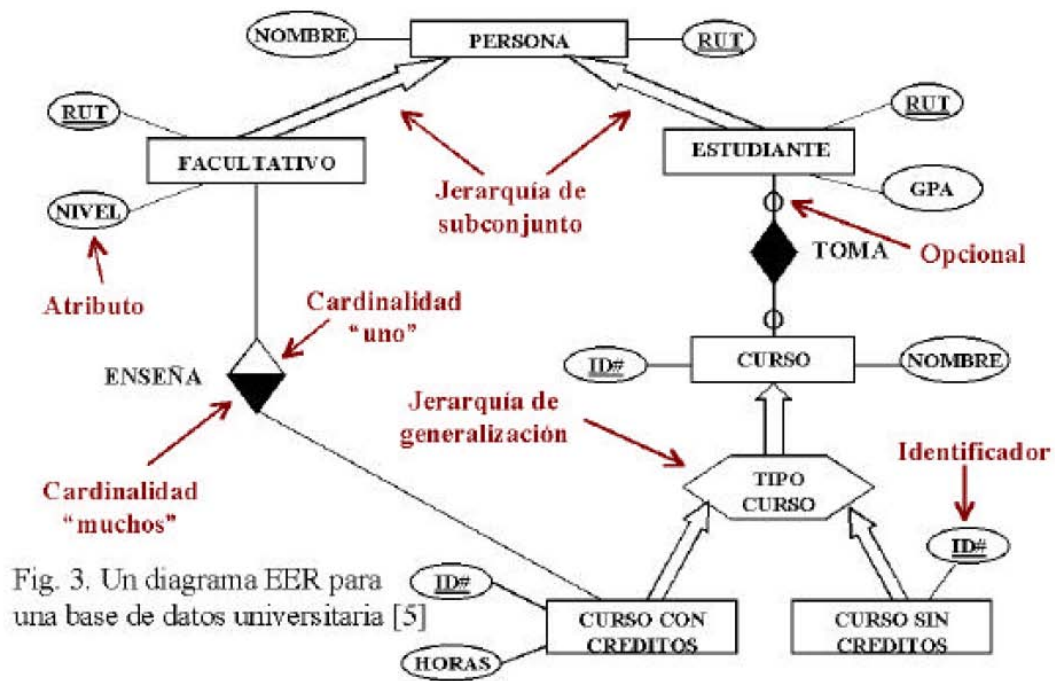


Fig. 3. Un diagrama EER para una base de datos universitaria [5]

Figura A.8 Ejemplo del Modelo ERE de Teorey, Yang y Fry

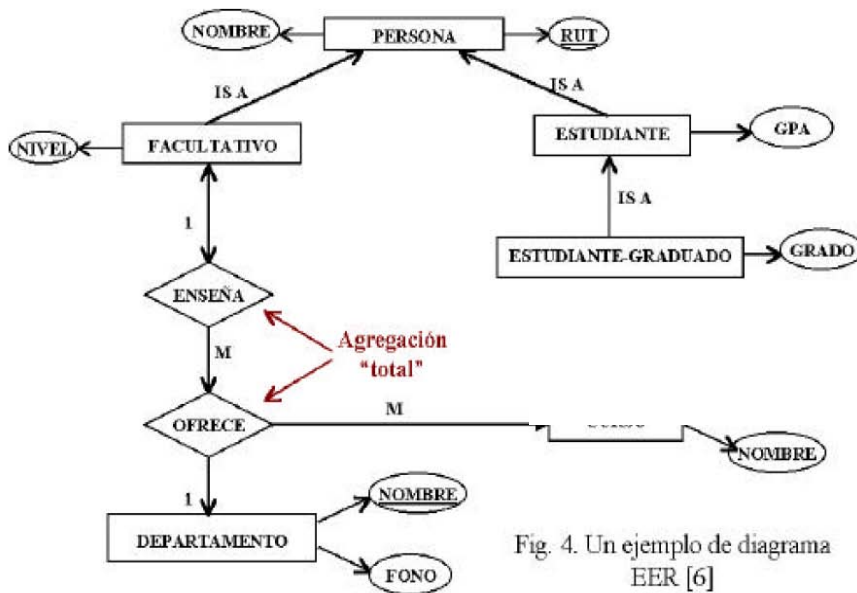


Fig. 4. Un ejemplo de diagrama EER [6]

Figura A.9 Ejemplo del Modelo ERE de Markowitz y Shoshani

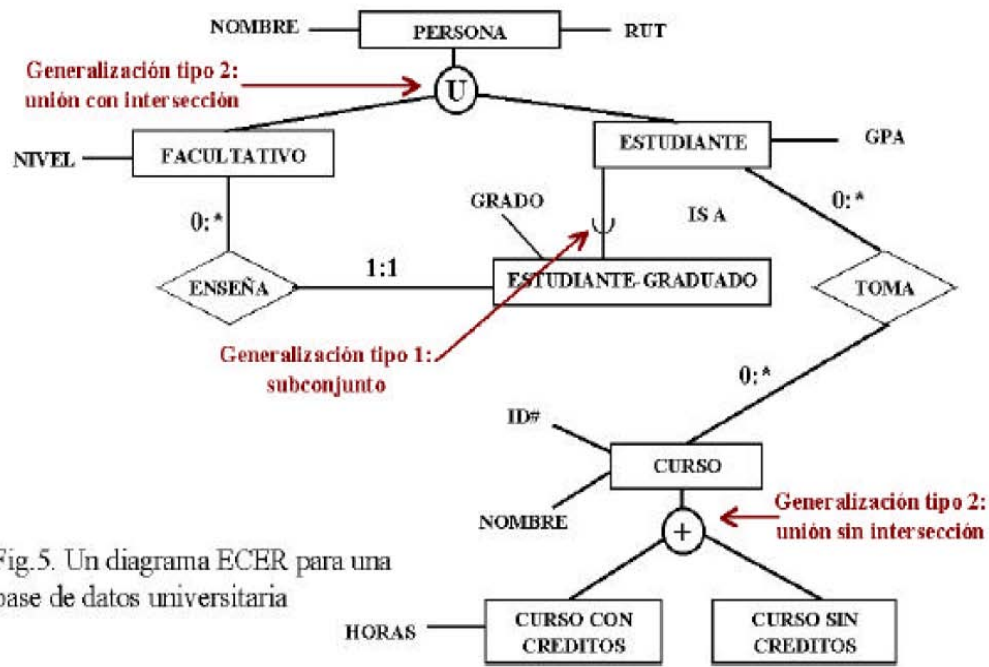


Fig. 5. Un diagrama ECR para una base de datos universitaria

Figura A.10 Ejemplo del Modelo Entidad Relación Conceptual Extendido (ECR)

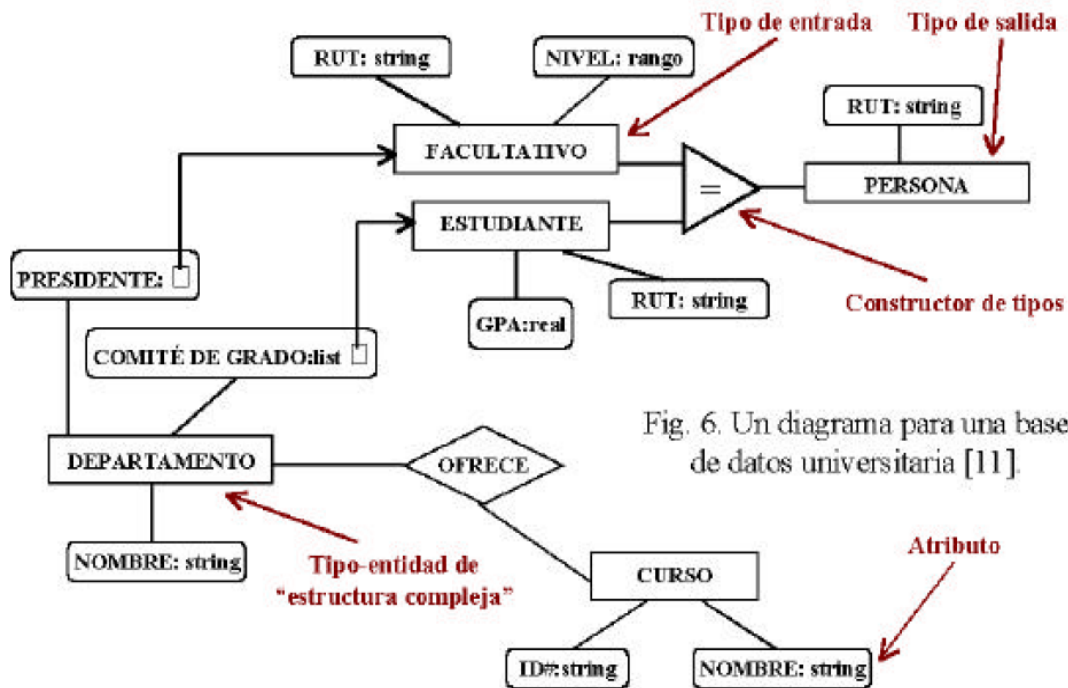


Fig. 6. Un diagrama para una base de datos universitaria [11].

Figura A.11 Ejemplo del Modelo ERE de Hohenstein y Gogolla.