

Contenido

I. Idea de Producto.

I.1. Identificación de Usuarios Participantes.

I.2. Planificación y Realización de Entrevistas.

I.3. Especificación de Requisitos.

I.3.1. Objetivos y Alcance del Sistema.

I.3.2. Definiciones, Acrónimos y Abreviaturas.

I.3.3. Descripción General.

I.3.4. Requisitos Funcionales.

I.3.5. Requisitos de Usuario y Tecnológicos.

I.3.6. Requisitos de Interfaces Externas.

I.3.7. Requisitos de Rendimiento.

I.3.8. Requisitos de Desarrollo y Restricciones de Diseño.

I.4. Estudio de Viabilidad.

I.4.1. Viabilidad Económica.

I.4.2. Viabilidad Técnica.

I.4.3. Viabilidad Operativa.

II. Especificación C.

II.1. Modelización del Sistema.

II.1.1. Diagramas de Clases.

II.1.2. Diagramas de Casos de Uso.

II.1.3. Diagramas de Transición de Estados.

II.2. Identificación de Roles.

II.3. Guiones y Escenarios.

II.3.1. Diagramas de Transición de Escenarios.

II.3.2. Tablas de Transición de Escenarios.

II.3.3. Descripción de Escenarios.

II.3.4. Fichas de Objetos de Escenario.

III. Especificación D.

III.1. Arquitectura Física del Sistema.

III.2. Fichas Técnicas de Métodos de Objetos.

III.3. Fichas Técnicas de Eventos de Objetos de Escenario.

III.4. Diseño de la Estructura de Datos.

III.4.1. Modelo Lógico del Sistema

III.4.2. Modelo Físico del Sistema

IV. Codificación

IV.1. Elección del Lenguaje

IV.2. Código Fuente

I. Idea de Producto

Este módulo tiene como objetivo analizar y documentar las necesidades funcionales que deberán ser soportadas por el sistema a desarrollar. Para ello, se identificarán los requisitos que ha de satisfacer el nuevo sistema mediante entrevistas, el estudio de los problemas de las unidades afectadas y sus necesidades actuales. Además de identificar los requisitos se deberán establecer las prioridades, lo cual proporciona un punto de referencia para validar el sistema final que compruebe que se ajusta a las necesidades del usuario.

I.1. Identificación de Usuarios Participantes.

Se identificaron los siguientes usuarios:

- **Usuario:** es la persona que usa la agenda

I.2. Planificación y Realización de Entrevistas.

En este apartado aparece la transcripción de las entrevistas con los usuarios.

I.3. Especificación de Requisitos.

El objetivo de la especificación es definir en forma clara, precisa, completa y verificable todas las funcionalidades y restricciones del sistema que se desea construir. Una vez aprobado, servirá de base al equipo para la construcción del nuevo sistema.

Esta especificación se ha realizado de acuerdo al estándar “IEEE Recommended Practice for Software Requirements Specifications (IEEE/ANSI 830-1993)”, y se basa en las entrevistas realizadas a los usuarios participantes y el estudio de la documentación existente.

I.3.1. Objetivos y Alcance del Sistema.

Los principales objetivos del sistema a desarrollar son la gestión de personas y sus datos en una agenda del tipo telefónico. El futuro sistema llevará el nombre de MiAgenda. Permitirá al usuario introducir, modificar y borrar información acerca de las personas, buscarlas, imprimir un listado del tipo agenda, avisar sobre sus cumpleaños y permitir llamarlas por teléfono mediante la conexión por MODEM a la red telefónica.

El desarrollo lo llevará a cabo (MiEmpresa), con opción a ser responsable del posterior mantenimiento del mismo.

I.3.2. Definiciones, Acrónimos y Abreviaturas.

Definiciones:

No se han definido

Acrónimos:

No se han definido

Abreviaturas:

SGBD: Sistema Gestor de Base de Datos

I.3.3. Descripción General.

Una agenda es típicamente una colección de información sobre personas relacionadas al dueño de la agenda, tales como: apellidos, nombres, fecha de nacimiento, dirección, teléfonos (particular, trabajo, fax, celular, etc.) y correo electrónico.

El sistema debe permitir al usuario crear nuevas personas, modificar su información y borrarlas. Debe permitir la emisión de un listado alfabético de personas, así como también debe mostrar un calendario con los diferentes cumpleaños. Una función a tener en cuenta es la posibilidad de marcado de un teléfono mediante la computadora, usando un MODEM.

I.3.4. Requisitos Funcionales.

Esta sección nos presenta una descripción general del sistema con el fin de conocer las funciones que debe soportar, los datos asociados, las restricciones impuestas y cualquier otro factor que pueda influir en la construcción del mismo.

Las funciones que debe realizar el sistema son las que se detallan a continuación:

- Gestión de personas: el sistema debe permitir introducir información acerca de las personas (apellidos, nombres, fecha de nacimiento, dirección, teléfonos y correo electrónico), modificarla y borrarla.
- Listado de personas: permite listar las personas de la agenda en orden alfabético.
- Consulta de calendario de cumpleaños: debe mostrar un calendario con las fechas de los cumpleaños de las personas.
- Verificación de cumpleaños: el sistema debe mostrar las personas que cumplen años en la fecha actual.
- Marcado telefónico: debe permitir marcar usando el MODEM de la computadora al teléfono seleccionado, mantener la comunicación y desconectarla.

I.3.5. Requisitos de Usuario y Tecnológicos.Requisitos de usuario:

El usuario será la persona dueña de la agenda. Las interfaces deben ser intuitivas, fáciles de usar y amigables.

Requisitos tecnológicos:

Se ha optado por un entorno económico y fácil de instalar. La aplicación se ejecutará sobre un esquema servidor de archivos. El sistema operativo será MS Windows 9x o bien MS Windows NT 4.0 Workstation. El Sistema Gestor de Base de Datos es MS Access 97.

I.3.6. Requisitos de Interfaces Externas.

Interfaces de usuario:

La interfaz de usuario debe ser orientada a ventanas tipo Windows.

Interfaces Hardware:

Ratón y teclado estándar.

Interfaces software:

No se han definido.

I.3.7. Requisitos de Rendimiento.

El tiempo de respuesta de la aplicación a cada función solicitada por el usuario no debe ser superior a los 5 segundos.

I.3.8. Requisitos de Desarrollo y Restricciones de Diseño.

Requisitos de Desarrollo:

El ciclo de vida será el de Prototipado Evolutivo, debiendo orientarse hacia el desarrollo de un sistema flexible que permita incorporar de manera sencilla cambios y nuevas funcionalidades.

Restricciones de Diseño:

- Ajuste a estándares: no se han definido.
- Seguridad: la seguridad de los datos será establecida por el Sistema Gestor de Base de Datos Relacional que se emplee.
- Política de Respaldo: no se ha definido.
- Base de Datos: el Sistema Gestor de Base de Datos debe ser relacional.
- Política de Borrado: no se ha definido.

I.4. Estudio de Viabilidad.

I.4.1. Viabilidad Económica:

Se debe definir un presupuesto estimado que incluye hardware, software y horas-hombre. Aparte se deben calcular los beneficios que se obtendrán con el nuevo sistema.

I.4.2. Viabilidad Técnica:

De acuerdo a los requisitos funcionales, determinar si es posible realizarlos con las tecnologías actuales.

I.4.3. Viabilidad Operativa:

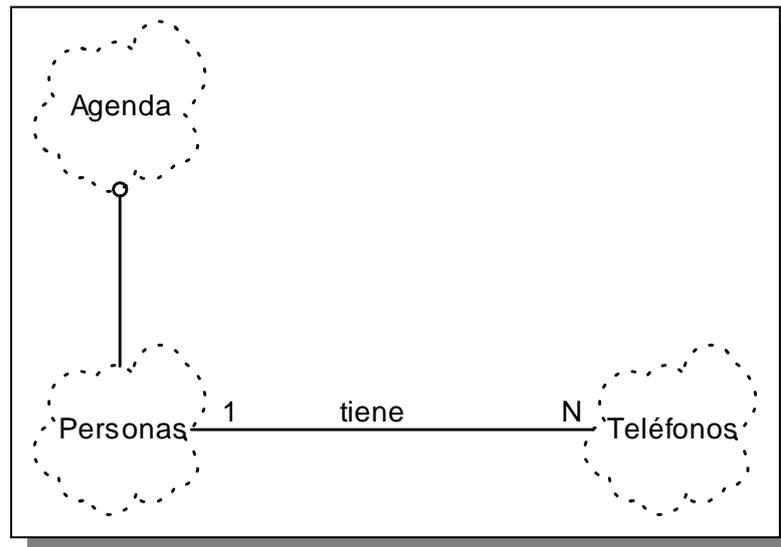
Determinar si se violan disposiciones legales, como ser Habeas Data.

II. Especificación C.

II.1. Modelización del Sistema.

II.1.1. Diagramas de Clases.

El diagrama de clases se muestra a continuación:



La especificación de clases se muestra a continuación:

Clase Personas	
Atributos	Operaciones
Apellidos: Cadena	Entero Edad()
Nombres: Cadena	Cadena LíneaTeléfonos()
FechaNacimiento: Fecha	Conjunto Teléfonos Teléfonos()
Dirección: Cadena	Cadena AgregarTeléfono(T: Teléfonos)
Email: Cadena	Cadena ModificarTeléfono(T: Teléfonos)
	Cadena QuitarTeléfono(T: Teléfonos)

Clase Teléfonos	
Atributos	Operaciones
TipoTeléfono: Cadena	Personas Persona()
Númeroteléfono: Cadena	Cadena Marcar()
	Cadena Colgar()

Clase Agenda	
Atributos	Operaciones
	Conjunto Personas Buscar (c: cadena, tipo: cadena)
	TotalPersonas()
	Conjunto Personas ListarPersonas()
	Cadena AgregarPersona(P: Personas)
	Cadena ModificarPersona(P: personas)
	Cadena QuitarPersona(P: personas)
	Conjunto Fechas Calendario()
	Cadena HoyCumple()

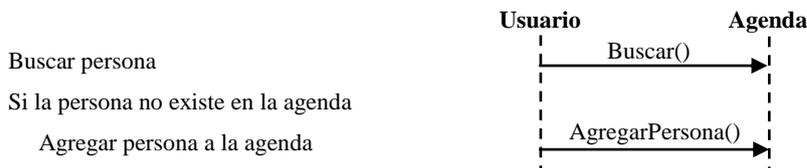
II.1.2. Diagramas de Casos de Uso.

Los casos de uso principales son:

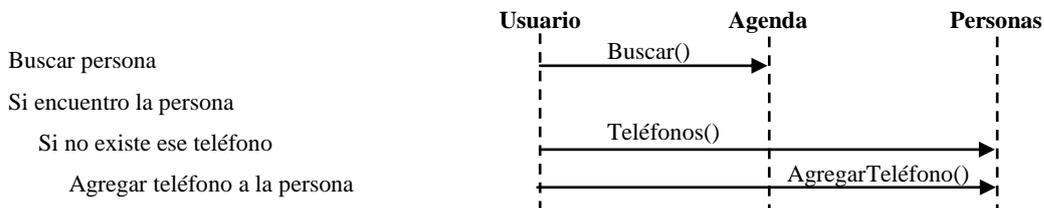
- a. Se desea agregar una persona a la agenda.
- b. Se desea agregar un teléfono a una persona.
- c. Se desea marcar un número de teléfono.
- d. Se desea imprimir un listado de personas.
- e. Se desea borrar una persona de la agenda.
- f. Se desea borrar un teléfono de una persona.
- g. Se desea ver el calendario de cumpleaños
- h. Se desea conocer quién cumple años en el día de la fecha.

Los diagramas de interacción pueden verse a continuación:

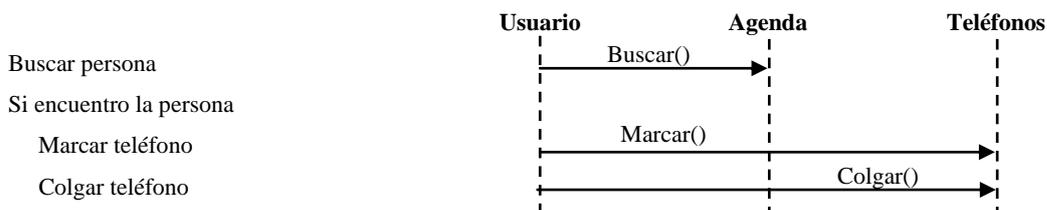
a. Se desea agregar una persona a la agenda.



b. Se desea agregar un teléfono a una persona.



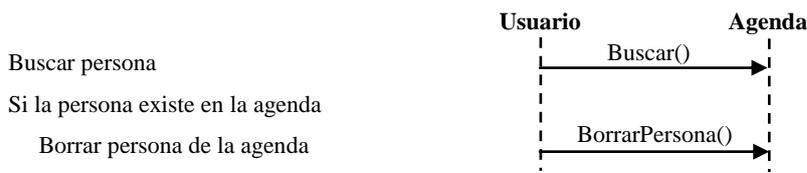
c. Se desea marcar un número de teléfono.



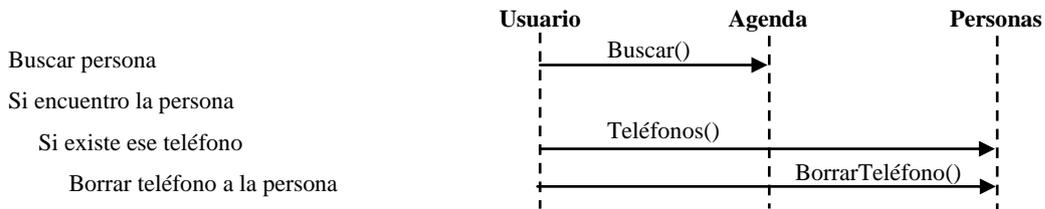
d. Se desea imprimir un listado de personas.



e. Se desea borrar una persona de la agenda.



f. Se desea borrar un teléfono de una persona.



g. Se desea ver el calendario de cumpleaños.

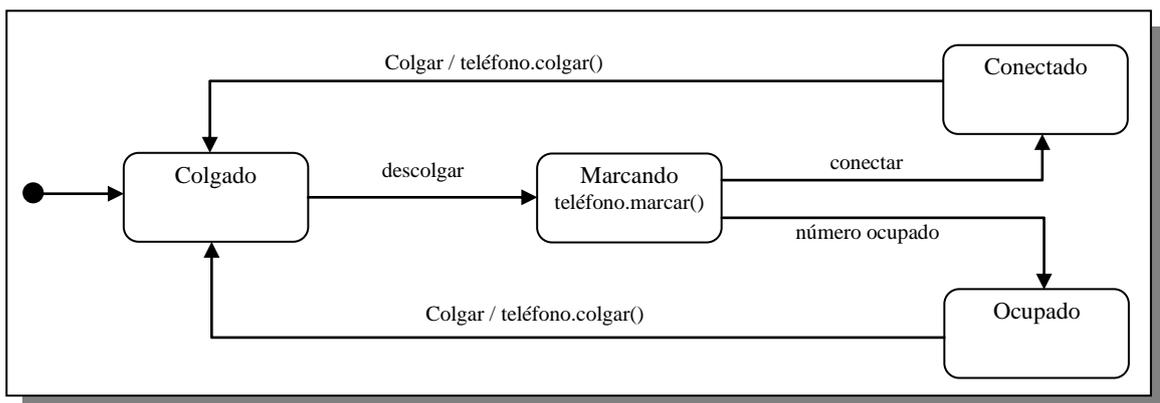


h. Se desea conocer quién cumple años en el día de la fecha.



II.1.3. Diagramas de Transición de Estados.

Se realiza el siguiente diagrama de transición de estados para la clase teléfono:



II.2. Identificación de Roles.

En esta etapa se pretende identificar los roles que cumplen los usuarios finales en el sistema, teniendo en cuenta las funciones que realizan y describiendo las restricciones a los datos de cada uno.

Rol Usuario:

Funciones que realiza:

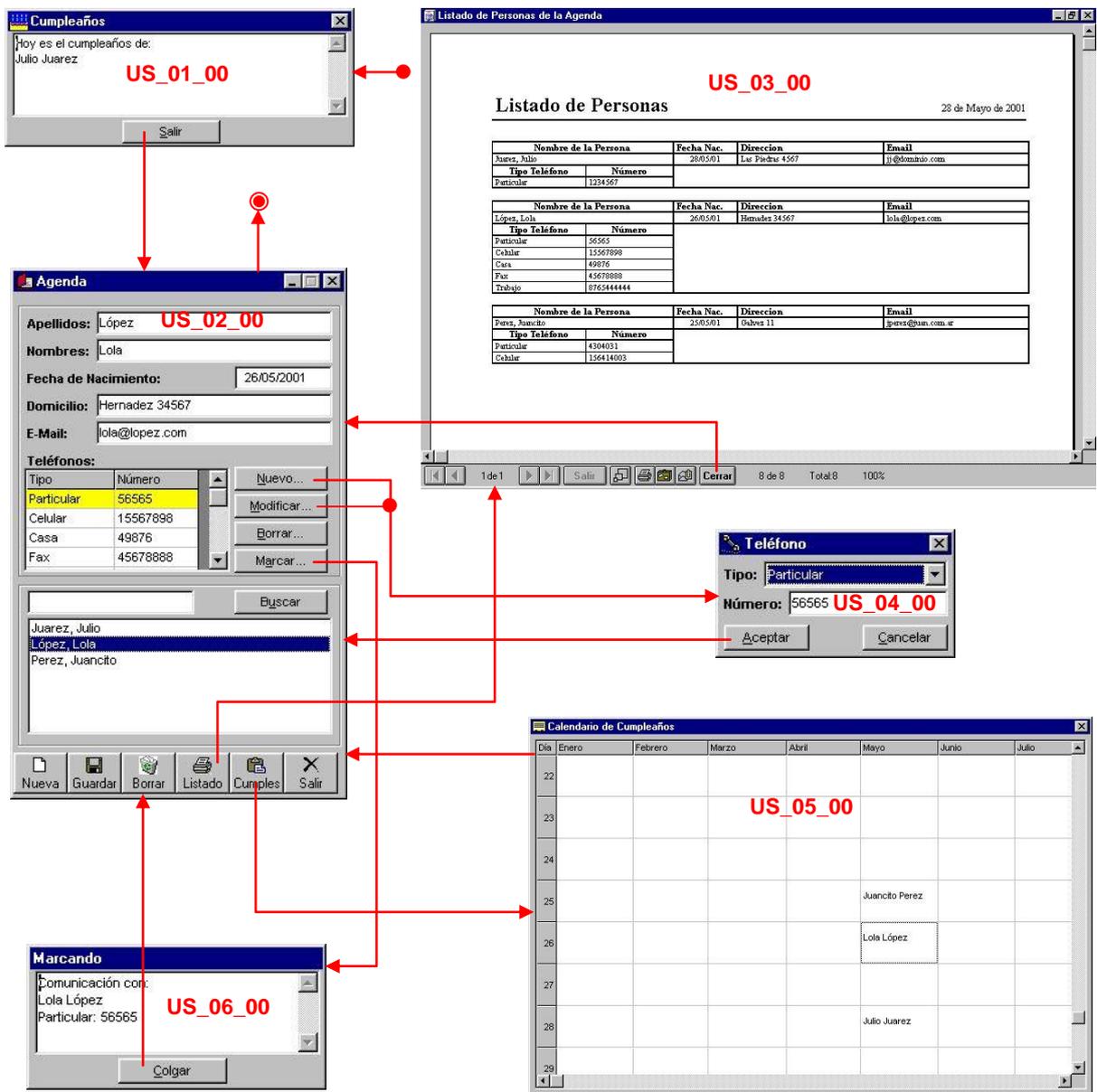
- Gestión de personas
- Impresión de listado de personas
- Marcado telefónico.
- Consulta de calendario de cumpleaños

Restricciones de datos: no se han definido

II.3. Guiones y Escenarios.

II.3.1. Diagramas de Transición de Escenarios.

Guión Usuario:



II.3.2. Tablas de Transición de Escenarios.

La tabla de transición de escenarios del guión usuarios se detalla a continuación:

	US_01_00	US_02_00	US_03_00	US_04_00	US_05_00	US_06_00
US_01_00	-----	US_01_02	-----	-----	-----	-----
US_02_00	-----	-----	US_02_25	US_02_14/5	US_02_26	US_02_17
US_03_00	-----	US_03_03/8	-----	-----	-----	-----
US_04_00	-----	US_04_05/6	-----	-----	-----	-----
US_05_00	-----	US_05_00	-----	-----	-----	-----
US_06_00	-----	US_06_02	-----	-----	-----	-----

II.3.3. Descripción de Escenarios.

Escenario US_01_00: Cumpleaños



Este es el escenario de inicio de la aplicación. Si es que existe alguna persona que cumple años en el día de la fecha, se carga la ventana que muestra en el cuadro de texto US_01_01 un listado de las personas que cumplen años. Al presionar el botón US_01_02, se descarga la ventana y se carga la ventana US_02_00.

Escenario US_02_00: Principal

Al cargarse la ventana se muestra en la lista US_02_21 todas las personas de la agenda en orden alfabético. En el registro de la persona se muestran los datos de la primera persona de la lista (apellidos, nombres, etc.). En la grilla US_02_13 se muestran los teléfonos de la persona. Los botones US_02_14 a US_02_17 se muestran sólo en el caso de una persona existente y se ocultan ante el ingreso de una nueva persona. El botón US_02_14 permite agregar un nuevo teléfono, por lo que carga el escenario US_04_00. El botón US_02_15 permite modificar un teléfono existente y también llama al escenario US_02_04. El botón US_02_16 permite borrar el teléfono elegido en la grilla mientras que el botón US_02_17 marca el número de teléfono seleccionado en la grilla (carga el escenario US_06_00). El botón US_02_20 permite buscar personas por apellido o parte de su apellido, que se carga en el cuadro de texto US_02_19. El resultado de la búsqueda se muestra en la grilla US_02_21. En caso de encontrarse algún registro, se muestra el primero de la lista; si no se encuentran registro, se crea uno nuevo (vaciándose los campos de registro). Al hacer clic sobre la lista, se traen al registro los datos de la persona seleccionada. El botón US_02_22 permite crear un nuevo registro, el US_02_23 guarda los datos de un registro nuevo o existente, el US_02_24 permite borrar el registro seleccionado de la lista, el US_02_25 muestra el listado (carga escenario US_03_00), el US_02_26

muestra el calendario de cumpleaños (carga escenario US_05_00) y el US_02_27 descarga la ventana y finaliza la aplicación.

US_02_00

01

02 Apellidos: López **03**

04 Nombres: Lola **05**

06 Fecha de Nacimiento: 26/05/2001 **07**

08 Domicilio: Hernandez 34567 **09**

10 E-Mail: lola@lopez.com **11**

12 Teléfonos:

Tipo	Número
Particular	56565
Celular	15567898
Casa 13	49876
Fax	45678888

14 Nuevo...
15 Modificar...
16 Borrar...
17 Marcar...

18

19 **20** Buscar

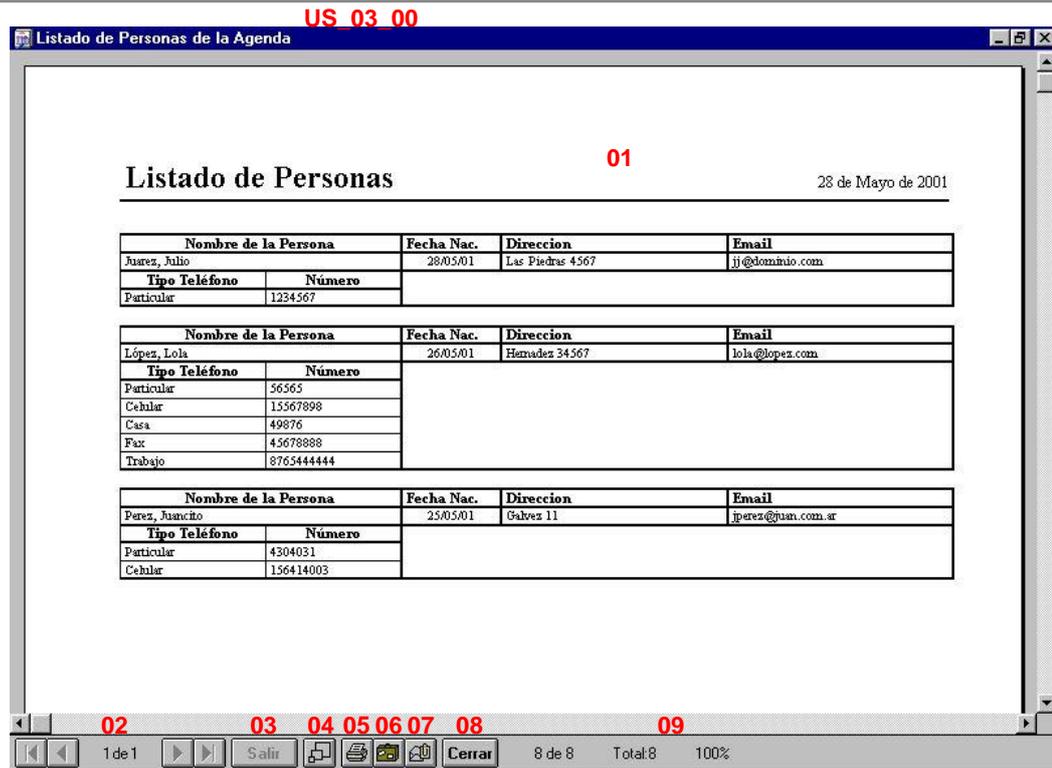
Juarez, Julio
López, Lola
Perez, Juancito

21

22 Nueva **23** Guardar **24** Borrar **25** Listado **26** Cumple **27** Salir

Escenario US_03_00: Listado

Este escenario permite visualizar en la página con opción a impresión, el listado de las personas, sus datos asociados y sus teléfonos. Al cargarse la ventana se muestra en la pantalla US_03_01 la presentación preliminar del listado de personas en orden alfabético, incluyendo todos sus teléfonos. La barra de botones US_03_03 permite navegar a través de las páginas, el botón US_03_03 descarga la ventana y vuelve al escenario US_02_00, el botón US_03_04 permite aumentar o disminuir la escala de presentación preliminar, el botón US_03_05 permite imprimir el listado en la impresora por defecto del sistema operativo, el botón US_03_06 permite importar el listado en diversos formatos (texto, html, xls, etc.), el botón US_03_07 permite adjuntar el listado a un correo electrónico usando el cliente de correo electrónico por defecto del sistema operativo y el botón US_03_08 sale de la ventana y vuelve a la US_02_00. La etiqueta de texto US_03_09 muestra los registros visualizados sobre el total de registros y su porcentaje.



Escenario US_04_00: Teléfono

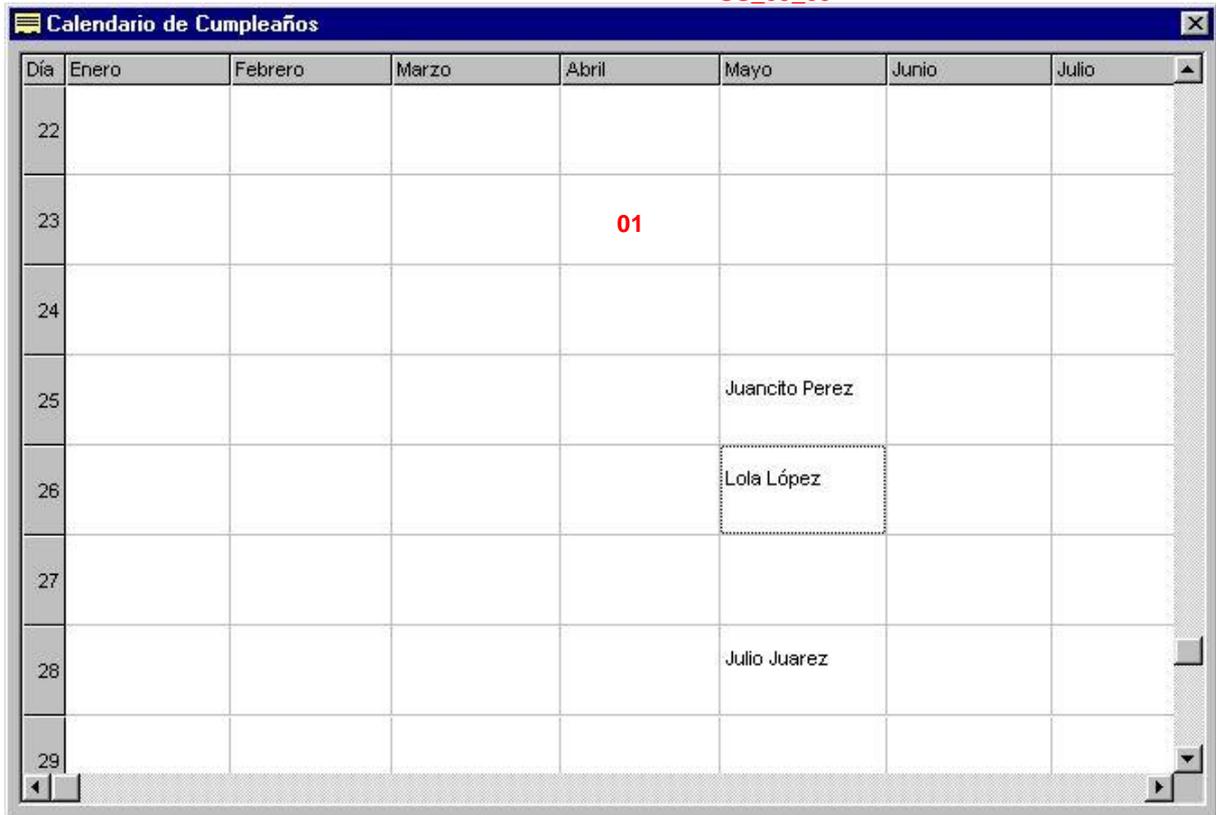


Esta ventana se carga en dos modalidades: alta y modificación. En la primera opción, se pone el título de la ventana como nuevo teléfono, se señala el primer elemento de la lista desplegable US_04_02 y se pone vacío el cuadro de texto US_04_04. En la segunda opción, se llenan los campos US_04_02 y US_04_04 con el tipo y número del teléfono que se quiere modificar. La lista desplegable US_04_02 tiene los tipos de teléfonos más usados (particular, casa, trabajo, celular y fax). Al presionar el botón US_04_05 (aceptar), se vuelve a la ventana US_02_00 con los datos del registro para ser guardado o modificado, mientras que si se presiona el botón US_04_06 (cancelar), se vuelve al escenario US_02_00 sin hacer nada.

Escenario US_05_00: Calendario

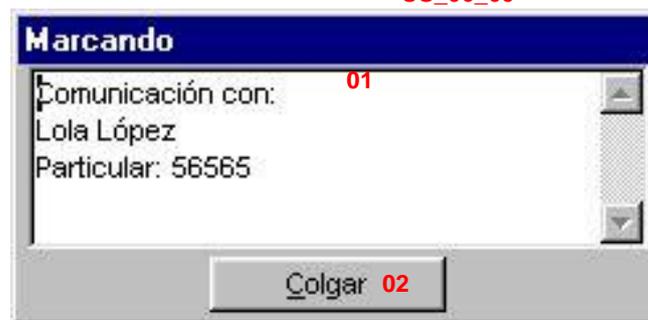
Al cargarse la ventana, se muestra en la grilla cuyas columnas son los meses y cuyas filas son los días, los nombres de las personas que cumplen años esa fecha. Al cerrar la ventana, se vuelve al escenario US_02_00.

US_05_00



Escenario US_06_00: Marcando

US_06_00



Al cargarse el escenario se muestra en el cuadro de texto US_06_01, la persona y el número al que se está llamando. Si no se puede establecer la comunicación, se avisa mediante un mensaje. En cualquier momento (establecida o no la comunicación) se puede colgar mediante el botón US_06_02, que vuelve al escenario US_02_00. Se debe escuchar por el parlante del equipo el establecimiento de la llamada.

II.3.4. Fichas de Objetos de Escenario.

La nomenclatura de identificación de escenarios y objetos se describe a continuación: dos letras que identifican al usuario al que pertenece el escenario; luego siguen dos números que identifican al escenario; finalmente dos números que identifican al objeto del escenario. Así el objeto US_02_21, será el objeto 21 del escenario 02 del guión usuario. El objeto 00 se reserva para el escenario propiamente dicho, que también es susceptible de interactuar con el usuario en forma de eventos.

Para la descripción de los objetos de entorno se adoptan estándares de tipos de objetos con similares características para facilitar la descripción. Estos estándares contienen las propiedades y los valores comunes a todos los objetos. Luego en la descripción de los mismos se procede a agregar las propiedades particulares de cada objeto.

Estándar de Ventanas

Estilo borde = 1 'No permitir cambiar tamaño

Tipo de letra

Nombre = "Arial"

Tamaño = 8

Subrayado = 0 'Falso

Itálica = 0 'Falso

Botón maximizar = 0 'Falso

Botón minimizar = 0 'Falso

Permitir movimiento = 1 'Verdadero

Estándar de Etiquetas

Tipo de letra

Nombre = "Arial"

Tamaño = 8

Subrayado = 0 'Falso

Itálica = 0 'Falso

Estándar de Máscaras de texto

Tipo de letra

Nombre = "Arial"

Tamaño = 8

Subrayado = 0 'Falso

Itálica = 0 'Falso

Estándar de Botones

Tipo de letra

Nombre = "Arial"

Tamaño = 8

Subrayado = 0 'Falso

Itálica = 0 'Falso

Altura = 375

Estándar de Cuadros de texto

Tipo de letra

Nombre = "Arial"

Tamaño = 8

Subrayado = 0 'Falso

Itálica = 0 'Falso

Luego, usando el lenguaje como herramienta CASE, se procede a listar todos los objetos de escenario de la siguiente manera:

<p>US_02_00 'Ventana Título = "Agenda" Altura = 2865 Posición inicial = 'Centrada en pantalla' Ancho = 5880</p>	<p>US_02_01 'Marco Altura = 1935 Posición izquierda = 120 Posición arriba = 0 Ancho = 5655</p>
<p>US_02_02 'Etiqueta Alineado = 1 Justificado derecha Título = "Apellidos:" Altura = 315 Posición izquierda = 240 Posición arriba = 360 Ancho = 1215</p>	<p>US_02_03 'Cuadro de texto Altura = 330 Posición izquierda = 1560 Texto ayuda = "Apellido del Alumno." Posición arriba = 360 Ancho = 1215 Valor inicial = ""</p>

Y así sucesivamente para cada uno de los objetos de escenario de todos los escenarios.

III. Especificación D.

III.1. Arquitectura Física del Sistema.

La arquitectura del sistema será de servidor de archivos, con el sistema gestor de base de datos y la aplicación ejecutándose en la misma computadora. Tanto la lógica de negocios como la lógica de interfaz residirá en la aplicación, facilitando el mantenimiento. Esta arquitectura presenta las siguientes ventajas.

- Rápida y fácil implantación.
- Bajo costo inicial.
- Acceso simple y dinámico a datos

Esta arquitectura no es recomendable para sistemas de más de un cliente y para un volumen elevado de información pues no es flexible, su mantenimiento es complejo y su rendimiento global es muy bajo.

Se debe tener en cuenta que los eventos sobre los objetos de entorno responden a los estándares del Sistema Operativo, y disparan procedimientos que son de carácter local a la ventana donde se producen. Eso no impide que modifiquen variables globales, ni convoquen procedimientos globales, ni soliciten servicios remotos del tipo de esquema cliente-servidor. La ventana pasa el control al procedimiento correspondiente al producirse el evento que, una vez concluido, devuelve el control a la ventana. Algunas veces los procedimientos convocan a otras ventanas o disparan eventos de la misma ventana, en cuyo caso se interrumpe la ejecución del procedimiento, se pasa el control a la ventana o evento correspondiente y luego de ejecutarse el/los procedimiento(s), vuelve el control al procedimiento que los convocó. Al descargarse una ventana, devuelve el control al procedimiento que la convocó.

III.2. Fichas Técnicas de Métodos de Objetos.

Clase Teléfonos

Método:	Personas Persona()
Conectarse con base de datos Traer registro de la persona con el IdPersona del teléfono Poner datos del registro en el objeto	

Método:	Cadena Marcar(m: modem)
Marcar = "OK" m.marcar número Si hay algún error ponerlo en Marcar como cadena	

Método:	Cadena Colgar(m: modem)
Colgar = "OK" m.colgar() Si hay algún error ponerlo en Colgar como cadena	

Clase Personas

Método:	Entero Edad()
Edad= Parte Entera (FechaActual – FechaNacimiento)	

Método:	Cadena LíneaTeléfonos
Conectarse con base de datos LíneaTeléfonos="" Para cada teléfono de la persona hacer LíneaTeléfonos = LíneaTeléfonos + teléfono.Tipo + ":" + teléfono.número + "-"	

Método:	Conjunto Teléfonos Teléfonos()
Conectarse con base de datos Teléfonos= vacío Para cada objeto teléfono de la persona hacer Agregar teléfono a Teléfonos	

Método:	Cadena AgregarTeléfono (T: Teléfonos)
Conectarse con base de datos AgregarTeléfono="OK" Si Número=vacío entonces AgregarTeléfono="Falta Número de teléfono" Sino Obtener el siguiente IdTeléfono Agregar registro a la tabla teléfonos con IdPersona	

Método:	Cadena ModificarTeléfono (T: Teléfonos)
Conectarse con base de datos ModificarTeléfono="OK" Si Número=vacío entonces ModificarTeléfono="Falta Número de teléfono" Sino Modificar el registro del teléfono en la tabla teléfonos	

Método:	Cadena QuitarTeléfono (T: Teléfonos)
Conectarse con base de datos QuitarTeléfono ="OK" Preguntar si quiere borrar teléfono Si la respuesta es afirmativa entonces Borrar registro de la tabla teléfonos	

Método:	Teléfonos DamePersona()
Conectarse con base de datos Traer registro de la tabla personas que coincida con IdPersona Poner datos del registro en la persona	

Clase Agendas

Método:	Conjunto Personas Buscar (c: cadena, t: cadena)
Conectarse con base de datos Buscar registros en la tabla personas cuyo apellido comience con c buscar = vacío Para cada persona encontrado hacer Agregar persona a Buscar	

Método: Entero TotalPersonas()
 Conectarse con base de datos
 TotalPersonas = Número de registros de la tabla personas

Método: Conjunto Personas ListarPersonas()
 Conectarse con base de datos
 Para cada persona de la tabla personas ordenada alfabéticamente por apellidos, nombres hacer
 Agregar persona a ListarPersonas

Método: Cadena AgregarPersona(p: Personas)
 Conectarse con base de datos
 AgregarPersona = "OK"
 Si Apellidos = vacío ó Nombres = vacío o FechaNacimiento incorrecta entonces
 AgregarPersona = "Error en los datos"
 Sino
 Obtener el siguiente IdPersona
 Agregar registro a la tabla personas

Método: Cadena ModificarPersona(p: Personas)
 Conectarse con base de datos
 ModificarPersona = "OK"
 Si Apellidos = vacío ó Nombres = vacío o FechaNacimiento incorrecta entonces
 ModificarPersona = "Error en los datos"
 Sino
 Modificar registro en la tabla personas

Método: Cadena QuitarPersona(p: Personas)
 Conectarse con base de datos
 QuitarPersona = "OK"
 Preguntar si quiere borrar persona
 Si la respuesta es afirmativa entonces
 Borrar registro de la tabla personas

Método: Conjunto Fechas Calendario()
 Conectarse con base de datos
 Para cada persona de la tabla personas hacer
 Agregar FechaNacimiento, Nombre a Fechas

Método: Cadena HoyCumple()
 Conectarse con base de datos
 HoyCumple = ""
 Para cada persona de la tabla personas cuya día y mes de nacimiento coincida con los actuales hacer
 HoyCumple = HoyCumple + Nombres + " " + Apellidos + salto_de_línea

III.3. Fichas Técnicas de Eventos de Objetos de Escenario.

Escenario US_01_00: Cumpleaños

Nombre:	US_01_00	Método	Abrir (c: Cadena)
Hacer US_01_01 = c Mostrar ventana US_01_00			

Nombre:	US_01_02	Evento	Click
Descargar ventana			

Escenario US_02_00: Principal

Nombre:	US_02_00	Evento	Carga de Ventana
Si Agenda.HoyCumple() <> "" entonces US_01_00.Abrir(Agenda.HoyCumple()) Llenar lista US_02_21 con Agenda.Buscar "", "" Si existen elementos en la lista entonces Mostrar datos en el registro del primer elemento de la lista Habilitar los botones US_02_14 al US_02_17 Sino Preparar datos para nuevo registro Deshabilitar los botones US_02_14 al US_02_17			

Nombre:	US_02_07	Evento	Salir de foco
Si el dato en US_02_07 no es una fecha entonces US_02_07 = FechaActual Poner foco en US_02_07			

Nombre:	US_02_13	Evento	Click
Si hay filas en la grilla Señalar la grilla elegida			

Nombre:	US_02_14	Evento	Click
US_04_00.Abrir ("A", "", "") Si no apreté el botón Cancelar entonces Pongo datos de US_04_00 en el objeto t del tipo Teléfonos Declaro p del tipo persona p.DamePersona() Si p.AgregarTeléfono(t) <> "OK" entonces Mostrar mensaje p.AgregarTeléfono(t) Sino Actualizar grilla US_02_13			

Nombre:	US_02_15	Evento	Click
US_04_00.Abrir ("M", t.tipo, t.número) Si no apreté el botón Cancelar entonces Declaro p del tipo persona p.DamePersona() Si p.ModificarTeléfono(t) <> "OK" entonces Mostrar mensaje p.ModificarTeléfono(t) Sino Actualizar grilla US_02_13			

Nombre:	US_02_16	Evento	Click
Declaro p del tipo persona p.DamePersona() Si p.QuitarTeléfono(t) <> "OK" entonces Mostrar mensaje p.QuitarTeléfono(t) Sino Actualizar grilla US_02_13			

Nombre:	US_02_17	Evento	Click
US_06_00.Abrir (p, t)			

Nombre:	US_02_19	Evento	Presionar Tecla
Si la tecla presionada es ENTER entonces Llenar lista US_02_21 con Agenda.Buscar US_02_19, "" Si existen elementos en la lista entonces Mostrar datos en el registro del primer elemento de la lista Habilitar los botones US_02_14 al US_02_17 Sino Preparar datos para nuevo registro Deshabilitar los botones US_02_14 al US_02_17			

Nombre:	US_02_20	Evento	Click
Llenar lista US_02_21 con Agenda.Buscar US_02_19, "" Si existen elementos en la lista entonces Mostrar datos en el registro del primer elemento de la lista Habilitar los botones US_02_14 al US_02_17 Sino Preparar datos para nuevo registro Deshabilitar los botones US_02_14 al US_02_17			

Nombre:	US_02_21	Evento	Click
Si la lista no está vacía entonces Mostrar datos del registro seleccionado en la lista Habilitar los botones US_02_14 al US_02_17			

Nombre:	US_02_22	Evento	Click
Preparar datos para nuevo registro Deshabilitar los botones US_02_14 al US_02_17			

Nombre:	US_02_23	Evento	Click
Poner los datos de la persona en p del tipo persona Si es un nuevo registro Si Agenda.AgregarPersona (p:persona) <> "OK" entonces Mostrar Mensaje Agenda.AgregarPersona (p:persona) Sino Mostrar datos del registro actual Habilitar los botones US_02_14 al US_02_17 Sino Si Agenda.ModificarPersona (p:persona) <> "OK" entonces Mostrar Mensaje Agenda.ModificarPersona (p:persona) Sino Mostrar datos del registro actual			

Nombre:	US_02_24	Evento	Click
Si Agenda.QuitarPersona (p:persona)<>"OK" entonces Mostrar Mensaje Agenda.QuitarPersona (p:persona) Sino Llenar lista US_02_21 con Agenda.Buscar US_02_19, "" Si existen elementos en la lista entonces Mostrar datos en el registro del primer elemento de la lista Habilitar los botones US_02_14 al US_02_17 Sino Preparar datos para nuevo registro Deshabilitar los botones US_02_14 al US_02_17			

Nombre:	US_02_25	Evento	Click
Cargar ventana US_03_00			

Nombre:	US_02_26	Evento	Click
Cargar ventana US_05_00			

Nombre:	US_02_26	Evento	Click
Descargar ventana US_02_00 Fin			

Escenario US_03_00: Listado

Nombre:	US_03_00	Evento	Carga de ventana
Conectar con base de datos Reporte US_03_01 = Agenda.ListarPersonas()			

Nombre:	US_03_03	Evento	Click
Descargar Ventana			

Nombre:	US_03_08	Evento	Click
Descargar Ventana			

Escenario US_04_00: Teléfono

Nombre:	US_04_00	Método	Abrir (m: cadena, t: cadena, n:cadena)
Según sea m m = "Alta": Mostrar primer elemento de US_04_02 US_04_04="" m = "Modificación" US_04_02 = t US_04_04 = n US_04_00.Cancelar = Verdadero Mostrar ventana			

Nombre:	US_04_05	Evento	Click
US_04_00.Cancelar = Falso US_04_00.Tipo = US_04_02 US_04_00.Numero = US_04_04			

Nombre:	US_04_06	Evento	Click
Cancelar = Verdadero Descargar ventana			

Escenario US_05_00: Calendario

Nombre:	US_05_00	Evento	Carga de ventana
Para cada E de Agenda.Calendario hacer US_05_01(Día(E.Fecha), Mes(E.Fecha)) = US_05_01(Día(E.Fecha), Mes(E.Fecha)) + E.Nombre			

Escenario US_06_00: Marcando

Nombre:	US_06_00	Método	Abrir (p: personas, t: teléfonos)
US_06_01 = “Comunicación con:” + p.nombres + “ “ + p.apellidos + t.tipo + “:” + t.numero Mostrar ventana			

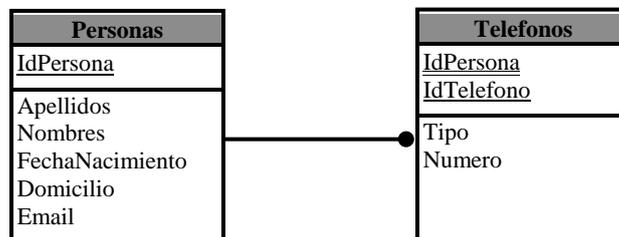
Nombre:	US_06_00	Evento	Carga de ventana
Si t.marcar<>”OK” entonces Mostrar mensaje t.marcar			

Nombre:	US_06_02	Evento	Click
Si t.colgar<>”OK” entonces Mostrar mensaje t.colgar Descargar ventana			

III.4. Diseño de la Estructura de Datos.

III.4.1. Modelo Lógico del Sistema

Se elige como modelo lógico el modelo relacional. El atributo o conjuntos de atributos que forman la clave primaria se encuentran con un subrayado simple. Los atributos que forman la clave foránea se encuentran con un subrayado doble.



III.4.2. Modelo Físico del Sistema

En esta actividad se define la estructura física de los datos utilizada en el sistema. Este esquema contempla las maneras de acceder físicamente a los datos, por medio de claves, índices, etc., para cumplir todos los requisitos.

En esta fase se tienen que tener en cuenta aspectos de funcionalidad y rendimiento en el funcionamiento de la aplicación. Es decir, se modificará el diseño lógico de datos, para convertirlo en un modelo físico, que contemple todas estos aspectos de funcionalidad y rendimiento. Se comentarán de manera individualiza el porque de cada una de las modificaciones realizadas en el modelo físico, justificando su ejecución.

Las tablas se describen usando el tipo de datos de MS Access:

Tabla Personas		
Columna	Tipo de datos	Permitir nulo
<u>IdPersona</u>	Entero Largo	No
Apellidos	Cadena (100)	No
Nombres	Cadena (100)	No
FechaNacimiento	Fecha	Si
Domicilio	Cadena (100)	Si
Email	Cadena (100)	Si

Tabla Telefonos		
Columna	Tipo de datos	Permitir nulo
<u>IdPersona</u>	Entero Largo	No
<u>IdTelefono</u>	Entero Largo	No
Tipo	Cadena (50)	No
Numero	Cadena (20)	No

Para optimizar las búsquedas se crean los siguientes índices.

- Un índice sin duplicados por tabla, formado por la clave primaria de la misma.
- Un índice sin duplicados por tabla, formado por la clave foránea de la misma.
- Un índice de la columna Apellidos de la tabla Personas sin duplicados.

IV. Codificación

IV.1. Elección del Lenguaje

Independientemente del paradigma de ingeniería del software, el lenguaje de programación tendrá impacto en la planificación, el análisis, el diseño, la codificación, la prueba y el mantenimiento de un proyecto. El lenguaje elegido para la construcción del producto es MS Visual Basic 5.0. Para construir los reportes se usará Seagate Crystal Reports 4.0 y para construir la base de datos se usará MS Access 97.

IV.2. Código Fuente

```

VERSION 1.0 CLASS
BEGIN
    MultiUse = -1 'True
END
Attribute VB_Name = "Agendas"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = True
Attribute VB_PredeclaredId = False
Attribute VB_Exposed = False
'Operaciones
Public Function buscar(db As Database, c As String, t As String) As Recordset
    Dim sql As String
    Dim rs As Recordset
    sql = "SELECT IdPersona, Apellidos + ', ' + Nombres as Nombre FROM Personas " & _
        "WHERE Apellidos LIKE '" & c & "'* ORDER By Apellidos,Nombres"
    Set rs = db.OpenRecordset(sql)
    Set buscar = rs
End Function

Public Function ListarPersonas(db As Database) As Recordset
    Dim sql As String
    Dim rs As Recordset
    sql = "SELECT * FROM Personas ORDER By Apellidos,Nombres"
    Set rs = db.OpenRecordset(sql)
    Set ListarPersonas = rs
End Function

Public Function AgregarPersona(db As Database, p As Persona) As String
    Dim er As error
    Dim texto As String
    Dim aux As String
    Dim sql As String
    Dim cod As Long
    Dim rs As Recordset
    AgregarPersona = "OK"
    If p.Apellidos = "" Then
        AgregarPersona = "No puede dejar vacío el apellido"
        Exit Function
    End If
    If p.Nombres = "" Then
        AgregarPersona = "No puede dejar vacío el nombre"
        Exit Function
    End If
    sql = "SELECT MAX(IdPersona) as Cod FROM Personas"
    Set rs = db.OpenRecordset(sql)
    If Not IsNull(rs("cod")) Then
        cod = rs("cod") + 1
    Else
        cod = 1
    End If
    rs.Close
    aux = Year(p.FechaNacimiento) & "/" & Month(p.FechaNacimiento) & "/" & Day(p.FechaNacimiento)
    sql = "INSERT INTO Personas VALUES(" & cod & ",'" & p.Apellidos & "','" & _
        p.Nombres & "','" & aux & "','" & p.Dirección & "','" & p.Email & "'"
    On Error GoTo error
    db.Execute sql
    On Error GoTo 0
    Exit Function
error:
    For Each er In Errors
        texto = texto & er.Description
    Next
    AgregarPersona = texto
End Function

Public Function ModificarPersona(db As Database, p As Persona) As String
    Dim er As error
    Dim aux As String

```

```

Dim texto As String
Dim sql As String
ModificarPersona = "OK"
If p.Apellidos = "" Then
    ModificarPersona = "No puede dejar vacío el apellido"
Exit Function
End If
If p.Nombres = "" Then
    ModificarPersona = "No puede dejar vacío el nombre"
Exit Function
End If
aux = Year(p.FechaNacimiento) & "/" & Month(p.FechaNacimiento) & "/" & Day(p.FechaNacimiento)
sql = "UPDATE Personas SET Apellidos='" & p.Apellidos & "', Nombres='" & _
    p.Nombres & "', FechaNacimiento=#" & aux & "#, " & _
    "Direccion='" & p.Dirección & "', Email='" & p.Email & "'" & _
    " WHERE IdPersona=" & p.IdPersona
On Error GoTo error
db.Execute sql
On Error GoTo 0
Exit Function
error:
For Each er In Errors
    texto = texto & er.Description
Next
ModificarPersona = texto
End Function

Public Function QuitarPersona(db As Database, p As Persona) As String
Dim er As error
Dim texto As String
Dim respuesta
Dim sql As String
QuitarPersona = "OK"
respuesta = MsgBox("¿Desea borrar a " & p.Nombres & " " & p.Apellidos & _
    "?", vbYesNo, "Advertencia")
If respuesta = vbNo Then Exit Function
On Error GoTo error
sql = "DELETE FROM Telefonos WHERE IdPersona=" & p.IdPersona
db.Execute sql
sql = "DELETE FROM Personas WHERE IdPersona=" & p.IdPersona
db.Execute sql
On Error GoTo 0
Exit Function
error:
For Each er In Errors
    texto = texto & er.Description
Next
QuitarPersona = texto
End Function

Public Function Calendario(db As Database) As Recordset
Dim rs As Recordset
Dim sql As String
sql = "SELECT MONTH(FechaNacimiento) as MES, DAY(FechaNacimiento) as Dia," & _
    "Nombres + ' ' + Apellidos as Nombre FROM Personas"
Set rs = db.OpenRecordset(sql)
Set Calendario = rs
End Function

Public Function HoyCumple(db As Database) As String
Dim m As Integer
Dim d As Integer
Dim sql As String
Dim rs As Recordset
Dim aux As String
m = Month(Date)
d = Day(Date)
sql = "SELECT Nombres + ' ' + Apellidos as Nombre FROM Personas" & _
    " Where Month(FechaNacimiento) =" & m & " And Day(FechaNacimiento) =" & d
Set rs = db.OpenRecordset(sql)
aux = ""
Do While Not rs.EOF
    aux = aux & rs("Nombre") & Chr(13) & Chr(10)
    rs.MoveNext
Loop
rs.Close
If aux <> "" Then aux = "Hoy es el cumpleaños de:" & Chr(13) & Chr(10) & aux
HoyCumple = aux
End Function

```