



UNIVERSIDAD NACIONAL AUTÓNOMA DE
MÉXICO

FACULTAD DE CIENCIAS

PROCESOS DE EXTRACCIÓN,
TRANSFORMACIÓN Y CARGA DE DATOS EN
SISTEMAS DE INFORMACIÓN

**REPORTE DE TRABAJO
PROFESIONAL**

QUE PARA OBTENER EL TÍTULO DE:
LICENCIADO EN CIENCIAS DE LA
COMPUTACIÓN

P R E S E N T A:

ALAN GUTIÉRREZ RUIZ

TUTOR:
DR. JAVIER GARCÍA GARCÍA

2014





Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

Hoja de datos del jurado

1. Datos del alumno

Gutiérrez

Ruiz

Alan

55 28 55 96

Universidad Nacional Autónoma de México

Facultad de Ciencias

Ciencias de la Computación

306159363

2. Datos del tutor

Dr

Javier

García

García

3. Datos del sinodal 1

Dra

Amparo

López

Gaona

4. Datos del sinodal 2

Dra

María de Luz

Gasca

Soto

5. Datos del sinodal 3

Dra

Hanna

Jadwiga

Oktaba

6. Datos del sinodal 4

M en C

Gustavo Arturo

Márquez

Flores

Índice

Lista de figuras	V
Lista de tablas	VI
Lista de algoritmos	VII
Introducción	1
Objetivo General	2
Justificación	3
1 Análisis de requerimientos	4
1.1 Objetivo	5
1.2 Requerimientos funcionales	5
1.2.1 Captura de Información Externa	6
1.2.2 Captura de Información Básica	17
1.2.3 Reportes	18
1.2.4 Administración de Catálogos	20
1.2.5 Administración del Sistema	20
1.3 Requerimientos no funcionales	21
1.4 Antecedentes del sistema	21
2 Diseño e implementación del sistema	23
2.1 Diseño del Sistema	23
2.2 Diagrama de base de datos y de data warehouse	29
2.3 Diseño de los ETL	45
2.4 Diseño del formulario con validación al vuelo	65
2.5 Diseño de la Administración de Catálogos y de Sistema	83

<i>ÍNDICE</i>	III
2.6 Arquitectura del Sistema	86
3 Resultados	89
4 Conclusiones y recomendaciones	91
A Carga de Información Externa	94
B Carga de Información Básica	95
C Administración de Catálogos del Sistema	98
D Exportación de Reportes	102
Referencias	123

Lista de Figuras

1.2	Ejemplar Fuente 1 de Información Externa, columnas	7
1.1	Ejemplar Fuente 1 de Información Externa	8
1.3	Ejemplar Fuente 2 de Información Externa	10
1.4	Ejemplar Fuente 2 de Información Externa, especificación	11
1.5	Ejemplar Fuente 3 de Información Externa	12
1.6	Ejemplar Fuente 4 de Información Externa	14
1.7	Ejemplar Fuente 5 de Información Externa	15
1.8	Ejemplar Fuente 6 de Información Externa	16
1.9	Entorno del cliente	19
2.1	Diagrama general de casos de uso	29
2.2	Diagrama de la base de datos, parte I	31
2.3	Diagrama de la base de datos, parte II	32
2.4	Diagrama del data warehouse, parte I	41
2.5	Diagrama del data warehouse, parte II	42
2.6	Diagrama del data warehouse, parte III	43
2.7	Diagrama del data warehouse, parte IV	44
2.8	Prototipo captura de Información Básica	66
2.9	Idea para autómeta de validación	68
2.10	Autómeta final para validación	70
2.11	Arquitectura del sistema	87
2.12	Arquitectura de ASP.NET MVC	88
A.1	Carga de Información Externa	94
B.1	Carga de Información Básica, validando la regla 6 en el día 1	96
B.2	Carga de Información Básica, validando la regla 9 en el día 2	97
C.1	Ejemplo Catálogo - Alta	98

C.2	Ejemplo Catálogo - Listado y Búsqueda	99
C.3	Ejemplo Catálogo - Consulta	99
C.4	Ejemplo Catálogo - Modificación	99
C.5	Ejemplo Catálogo - Baja	100
C.6	Ejemplo de validaciones en Catálogos	101
D.1	Monitoreo de Organismos de Alojamiento, Exportación de datos a hojas de cálculo	103
D.2	Monitoreo de Organismos de Alojamiento, Exportación de datos a hojas de cálculo	104
D.3	Ejemplo de Reportes - Actividad Aeroportuaria	105
D.4	Ejemplo de Reportes - Actividad Aeroportuaria	106
D.5	Ejemplo de Reportes - Actividad Aeroportuaria	107
D.6	Ejemplo de Reportes - Actividad Aeroportuaria	108
D.7	Ejemplo de Reportes - Conectividad Aérea, Servicio Doméstico	109
D.8	Ejemplo de Reportes - Conectividad Aérea, Servicio Doméstico	110
D.9	Ejemplo de Reportes - Conectividad Aérea, Servicio Internacional . .	111
D.10	Ejemplo de Reportes - Conectividad Aérea, Servicio Internacional . .	112
D.11	Ejemplo de Reportes - Movimiento de Cruceros	113
D.12	Ejemplo de Reportes - Movimiento de Cruceros	114
D.13	Ejemplo de Reportes - Llegada de Visitantes por Nacionalidad	115
D.14	Ejemplo de Reportes - Conectividad Aérea (asientos)	116
D.15	Ejemplo de Reportes - Conectividad Aérea (frecuencia)	117
D.16	Ejemplo de Reportes - Llegada de Visitantes por Nacionalidad	118
D.17	Ejemplo de Reportes - Oferta de Alojamiento por Estado	119
D.18	Ejemplo de Reportes - Viajeros Internacionales a México	120
D.19	Ejemplo de Reportes - Visitantes a Zonas Arqueológicas y Museos . .	121
D.20	Ejemplo de Reportes - Visitantes a Zonas Arqueológicas y Museos (anualizado)	122

Lista de Tablas

2.1	OLTP vs OLAP	30
2.2	Caso de uso Carga de Información Externa	48
2.3	Excepciones Caso de uso Carga de Información Externa	49
2.4	Reglas de validación en la captura de Información Básica	65
2.5	Caso de uso Carga de Información Básica	72
2.6	Excepciones Caso de uso Carga de Información Básica	74

Lista de Algoritmos

1	etlFuente1	50
2	etlFuente2	53
3	etlFuente3	55
4	etlFuente4	57
5	etlFuente5	60
6	etlFuente6	63
7	validaCuestionario	75
8	esNumeroEnteroPositivo	76
9	validaCelda	76
10	validaDia	77
11	validaRegla1	78
12	validaRegla2	79
13	validaRegla3	79
14	validaRegla4	80
15	validaRegla5	80
16	validaRegla7	81
17	validaRegla6	81
18	validaRegla8	82
19	validaRegla9	82
20	validaRegla10	83

Introducción

El presente trabajo es el producto de la experiencia profesional obtenida durante mi desempeño en el desarrollo de sistemas de software y, en particular, en la elaboración de un sistema computacional para una compañía que requería obtener información valiosa a partir de las grandes cantidades de datos que posee. Se trata de un proyecto privado, por lo que se omitirán ciertos detalles con el fin de respetar la confidencialidad del cliente.

El principal objetivo del sistema desarrollado fue que, a través de un módulo de procesamiento analítico de datos en línea (OLAP, por sus siglas en inglés), se debía obtener información agregada útil para la toma de decisiones. Entre otras características, el sistema debió concebirse de tal forma que fuese capaz de identificar y resolver las discrepancias que la integración de distintas fuentes heterogéneas de datos pudiera ocasionar. Por otra parte, se requirió que el sistema contase con un módulo específico de validación de datos *al vuelo*, que guiase a los usuarios de manera oportuna y eficaz en su proceso de captura de datos. Finalmente, para tener un completo control acerca del sistema, se debió contar con un módulo de seguridad y de autenticación de usuarios, así como uno de administración de catálogos de sistema.

En los siguientes capítulos se expondrán los antecedentes del proyecto y el planteamiento, tanto general como específico, de la problemática a resolver. Luego, se mostrará el análisis, estrategia y diseño de las herramientas necesarias para resolver el problema: se verán los procesos de ETL¹ que solucionaron la obtención de un único sistema para el análisis de datos por medio de OLAP; se presentará el diseño del autómata finito y de su algoritmo para la validación *al vuelo* en el módulo de captación de datos; y, se mostrará el desarrollo del administración de catálogos. Por último se presentarán los resultados obtenidos y conclusiones.

¹Extracción, Transformación y Carga de Datos (ETL, por sus siglas en inglés)

Objetivo general

Presentar un trabajo como producto de conocimientos y experiencias profesionales, el cual está basado en un proyecto de desarrollo de software a fin de mostrar las actividades de análisis, diseño, desarrollo e implementación que realicé en los módulos de ETL, captación de datos, y de administración de catálogos y del sistema.

Justificación

El caso de estudio está basado en un proyecto realizado, durante el periodo de Mayo de 2013 a Marzo de 2014, por parte de la empresa de desarrollo de software Pascola Comercial y Servicios SA de CV. Se trató de un proyecto de software, hecho a la medida, para una organización dedicada al turismo a nivel nacional.

El proyecto estuvo a cargo de un equipo de desarrollo de software, mi participación en el desarrollo del sistema fue como analista e ingeniero de software, responsable de hacer levantamiento de requerimientos, diseño de componentes, implementación de módulos, seguimiento de pruebas y errores, así como de la documentación.

Este trabajo respondió a la necesidad que tenía el cliente de contar con un sistema que le permitiera reunir, procesar y difundir datos acerca de las actividades económicas vinculadas al turismo en México. La información obtenida acerca de estos datos debía apoyar la toma de decisiones por parte del área ejecutiva de la empresa cliente.

Capítulo 1

Análisis de requerimientos

El cliente es una organización dedicada al turismo a nivel nacional. El ambiente en que se desarrollan sus actividades es la siguiente:

- El cliente reúne datos alrededor de la República Mexicana.
- Cuenta con múltiples organismos, dedicados o relacionados a la actividad turística, que generan datos de forma permanente.
- Los organismos pueden ser de servicios aeroportuarios, de cruceros, de alojamiento, de recreación, culturales, de administración migratoria y de control monetario.
- Los datos que se obtienen:
 - En servicios aeroportuarios: el número de vuelos de llegada y de salida, si se trata de vuelos nacionales o extranjeros, tipos de vuelo.
 - En servicios de alojamiento: el número de cuartos disponibles, de cuartos ocupados y si se trata de llegada de turistas nacionales o extranjeros.
 - En servicios de cruceros: el número de arribos de cruceros, así como, su número de pasajeros.
 - En servicios de recreación o culturales: el número de visitantes nacionales o extranjeros.
 - En servicios de administración migratoria: vuelos con destino o salida a México, con sus respectivos datos tales como número de vuelo, aerolínea, países y aeropuertos relacionados, número de asientos, frecuencia y escalas.

- En servicios de control monetario: ingresos y egresos (en millones de dólares) relacionados a los turistas nacionales e internacionales, en correspondencia con la cantidad de turistas registrados.
- Se tienen usuarios a nivel local, municipal o estatal que se encargan de generar y validar estos datos ya sea de forma semanal o mensual.
- Se tiene una base administrativa central que se encarga de obtener información acerca de los datos recopilados.
- El uso de la información obtenida es parte del *objetivo* del cliente.

1.1 Objetivo

El objetivo fue desarrollar un sistema de información estadística que recabe, procese y difunda información de las actividades económicas vinculadas al turismo en México.

1.2 Requerimientos funcionales

Los requerimientos funcionales fueron los siguientes:

- Captura de Información Externa
Se debía poder recolectar y subir datos al sistema, mediante archivos de hojas de cálculo (.xls) o de bases de datos (.mdb) para los datos provenientes de organismos de servicios aeroportuarios, de cruceros, de recreación, culturales, de administración migratoria y de control monetario.
- Captura de Información Básica
Se debía poder capturar datos provenientes de organismos de alojamiento, mediante formularios vía web.
- Reportes
Los datos capturados debían ser procesados y posteriormente explotados, de tal manera que fuesen de utilidad para el cliente.
- Administración de Catálogos
Se debía contar con la funcionalidad de almacenar y administrar las entidades que interactúan en las reglas de negocio del cliente.

- Administración del Sistema
Mediante niveles de seguridad el sistema debía permitir el correcto acceso a funciones y módulos destinados para su operación.

1.2.1 Captura de Información Externa

Los datos a cargar son proporcionados al cliente por parte de organismos de servicios aeroportuarios, de cruceros, de recreación, culturales, de administración migratoria y de control monetario. Dichos datos provienen de archivos que son enviados¹ al cliente en formato de hojas de cálculo (.xls) o de bases de datos (.mdb).

El sistema debía presentar una interfaz gráfica que permitiese a los usuarios automatizar sus procesos de recopilación de datos provenientes de seis fuentes de datos. A continuación se presentan cada una de las seis fuentes de datos con sus respectivas imágenes de ejemplo y observaciones que me permitieron identificar los elementos a cargar por medio del ETL.

Fuente 1

Son datos acerca de salidas y llegadas en aeropuertos mexicanos, vienen en un archivo de hoja de cálculo.

- El archivo consta de una sola pestaña.
- Hay una celda título (hasta arriba) que indica el mes y el año (período).
- Hay datos para un número variable de filas y veintitrés columnas:
 - Las primeras dos columnas indican año y mes (siempre es el mismo período que el de la celda título).
 - Las columnas 3 y 4 indican la sigla y el nombre de los organismos.
 - Las siguientes diecinueve columnas, de la 5 a la 23, son datos acerca de las actividades de los organismos para ese período.
 - Las últimas tres columnas indican datos totales:
 - * La columna 21 es la suma de las columnas 5,7,9,11,13,15,17 y 19.
 - * La columna 22 es la suma de las columnas 6,8,10,12,14,16,18 y 20.

¹El cliente no establece en qué medio le hacen llegar los archivos.

- * La columna 23 es la suma de las columnas 21 y 22.
- Para indicar que no hay valor en las actividades, se coloca un “-“
- Al final del documento hay una fila que contiene la suma total por cada columna.

En la Figura 1.1 se puede apreciar los encabezados, los renglones de datos y, al final, la sección de totales.

Existe una agrupación de la columnas 5 a la 20, dependiendo el concepto de *tipo de operación*. En la Figura 1.2 se puede ver que, además del Tipo de Operación, se tienen los conceptos de Nacional llegadas, Nacional salidas, Internacional llegadas e Internacional salidas.

- Las columnas 5,6,7,8 representan datos acerca del tipo de operación “CR”.
- Las columnas 9,10,11,12 representan datos acerca del tipo de operación “C”.
- Las columnas 13,14,15,16 representan datos acerca del tipo de operación “NR”.
- Las columnas 17,18,19,20 representan datos acerca del tipo de operación “G”.

Tipo de Operación			
Nacional		Internacional	
Llegadas	Salidas	Llegadas	Salidas

Figura 1.2: Ejemplar Fuente 1 de Información Externa, columnas

Fuente 2

Son datos acerca de ingresos y egresos de turistas en México, vienen en un archivo de hoja de cálculo.

- El archivo consta de una sola pestaña.
- Hay una celda título (hasta arriba) que indica el mes y el año (período).
- Se indican datos importantes para el cliente por medio de conceptos y su valor para ese período.
- Los datos están separados en distintos bloques:

Agosto de 2013																							
		Sigla	Aeropuerto	Comercial Regular				Charter				Comercial No Regular				General				Total		Total	
				Nacional		Internacional		Nacional		Internacional		Nacional		Internacional		Nacional		Internacional		Lleg.	Sal.		
				Lleg.	Sal.	Lleg.	Sal.	Lleg.	Sal.	Lleg.	Sal.	Lleg.	Sal.	Lleg.	Sal.	Lleg.	Sal.	Lleg.	Sal.	Lleg.	Sal.		
6	2013	Agosto	ACA	Acapulco	23,313	23,021	698	2,540	0	0	0	0	482	458	16	0	698	595	22	19	25,235	26,633	51,868
7	2013	Agosto	AGU	Aguascalientes	13,971	14,633	5,926	8,158	0	0	0	0	125	115	10	4	234	216	41	36	20,307	23,162	43,469
8	2013	Agosto	BJX	Bajío (León,Gto)	27,513	27,198	13,012	19,330	0	0	0	0	287	285	44	35	439	397	82	84	41,377	47,329	88,706
9	2013	Agosto	CJS	Cd. Juárez	34,275	30,020	0	0	847	505	0	0	64	75	0	2	412	453	21	79	35,619	31,134	66,753
10	2013	Agosto	CUN	Canón	241,416	257,848	367,894	411,806	26,724	29,044	44,856	46,183	2,102	1,820	30	56	348	396	272	310	683,642	747,463	1,431,105
11	2013	Agosto	CVJ	Cuernavaca	1,030	957	0	0	0	0	0	0	237	130	2	4	57	59	4	11	1,330	1,161	2,491
12	2013	Agosto	GDL	Guadalajara	255,879	257,456	111,125	139,682	0	0	0	0	691	887	93	110	591	996	113	116	368,492	399,247	767,739
13	2013	Agosto	HMD	Hermosillo	53,653	48,635	2,695	3,439	0	0	0	0	1,119	986	2	6	862	1,003	105	148	58,436	54,217	112,653
14	2013	Agosto	MEX	México	951,517	945,085	502,767	484,789	0	0	0	0	0	0	0	0	0	0	0	0	1,454,284	1,429,874	2,884,158
15	2013	Agosto	MID	Mérida	50,163	49,244	3,643	6,678	16	16	536	343	220	167	11	28	222	258	34	45	54,845	56,779	111,624
16	2013	Agosto	MXL	Mexicali	26,303	19,798	0	136	0	0	0	0	29	32	0	3	116	120	47	26	26,495	20,115	46,310
17	2013	Agosto	NLD	Nuevo Laredo	2,654	2,282	0	0	0	0	0	0	7	7	0	0	349	220	3	1	3,013	2,510	5,523
18	2013	Agosto	OAX	Oaxaca	20,141	18,924	1,071	5,503	0	0	0	0	649	643	6	6	348	312	3	0	22,218	25,388	47,606
19	2013	Agosto	PBC	Puebla	12,005	11,959	2,056	2,198	0	41	0	0	289	315	41	15	322	295	37	27	14,750	14,850	29,600
20	2013	Agosto	PVR	Puerto Vallarta	42,234	44,489	42,251	49,566	127	245	520	727	651	585	127	153	488	553	9	12	86,407	96,330	182,737
21	2013	Agosto	TAM	Tampico	23,913	22,208	1,483	2,597	0	0	0	0	537	571	8	9	776	779	134	140	26,851	26,304	53,155
22	2013	Agosto	TLC	Toluca	54,267	52,861	7,072	7,757	65	105	8	0	759	1,668	436	633	456	841	251	412	63,314	64,277	127,591
23	2013	Agosto	VER	Veracruz	39,709	37,573	3,404	5,724	151	146	98	0	442	436	46	20	308	257	35	38	44,193	44,194	88,387
24	2013	Agosto	VSA	Villahermosa	41,311	40,333	1,610	4,152	0	0	160	0	259	253	0	12	199	160	18	37	43,557	44,947	88,504
25	2013	Agosto	ZCL	Zacatecas	9,047	9,309	3,393	4,907	0	0	0	0	73	61	0	0	146	151	24	29	12,683	14,457	27,140
26	2013	Agosto	ZIH	Zihuatanejo	13,169	13,781	2,393	3,759	0	0	0	0	178	180	5	13	166	184	0	10	15,911	17,927	33,838
27																							
28				TOTAL RED	1,937,489	1,927,614	1,072,493	1,162,721	27,930	30,102	46,178	47,253	9,200	9,674	877	1,109	7,537	8,245	1,255	1,580	3,102,959	3,188,298	6,291,257
29																							

Figura 1.1: Ejemplar Fuente 1 de Información Externa

- Hay seis bloques de doce filas y dos columnas.
- En los bloques, la primera columna es para el nombre del concepto y la segunda para el valor de ese concepto.
- De arriba abajo, los cuatro primeros bloques traen la siguiente estructura:
 - * La primera fila trae la sumatoria del total del bloque.
 - * Contiene seis filas con datos primarios, es decir, a partir de ellos, se pueden calcular los demás datos en el bloque. Las seis filas son las 4,5,7,8,11,12 de arriba a abajo.
- Los dos bloques de hasta abajo traen la siguiente estructura:
 - * Estos dos bloques son derivados de los cuatro anteriormente descritos.
 - * Los valores de los conceptos del último bloque lado izquierdo se calculan a partir de los dos primeros bloques del lado izquierdo. Análogamente para el lado derecho.
 - * Para cada fila, sea V_1 y V_2 el valor de los bloques anteriores en esa misma fila. El nuevo valor es el producto de la multiplicación de 1000 por el cociente de la división de V_1 entre V_2 . (Es decir, $V_{i3} = 1000 * (V_{i1}/V_{i2})$)

En la Figura 1.3 se puede ver la imagen del archivo con los seis bloques. Los dos de hasta abajo son calculados a partir de los cuatro de arriba.

En la Figura 1.4 se puede ver la imagen de un bloque con sus respectivas relaciones. La primera columna indica el concepto, la segunda los datos, y la tercera es una columna adicional (sólo para aclarar las relaciones que hay entre las filas). Cabe aclarar que los conceptos presentados no son los mismos para todos los bloques.

Fuente 3

Son datos acerca de arribos de cruceros en puertos mexicanos, vienen en un archivo de hoja de cálculo.

- El archivo consta de una sola pestaña.
- Hay dos celdas título (hasta arriba) que indican el mes y el año (período).
- Hay datos separados en dos bloques:
 - Se indican datos importantes para el cliente por medio de bloques. Tienen un número constante de filas y tres columnas.

	A	B	C	D	E
1	2013				
2					
3	CONCEPTO	ago-13		CONCEPTO	ago-13
4					
5	SALDO (MILLONES DE DOLARES)	189.7			
6					
7	INGRESOS (MILLONES DE DOLARES)	1,030.9		EGRESOS (MILLONES DE DOLARES)	841.2
8					
9	Turistas internacionales	859.4		Turistas internacionales	577.8
10	Turistas de internación	801.5		Turistas de internación	552.8
11	Vía aérea	759.9		Vía aérea	417.8
12	Vía terrestre	41.6		Vía terrestre	135.0
13	Turistas fronterizos	57.9		Turistas fronterizos	25.0
14	Peatones	10.3		Peatones	2.8
15	En automóviles	47.6		En automóviles	22.2
16					
17	Excursionistas internacionales	171.5		Excursionistas internacionales	263.4
18	Excursionistas fronterizos	150.6		Excursionistas fronterizos	263.4
19	Peatones	29.7		Peatones	56.9
20	En automóviles	120.9		En automóviles	206.5
21					
22	NUMERO DE VIAJEROS (MILES)	6,541.6		NUMERO DE VIAJEROS (MILES)	7,807.1
23					
24	Turistas internacionales	2,079.2		Turistas internacionales	1,550.8
25	Turistas de internación	1,041.1		Turistas de internación	1,077.4
26	Vía aérea	907.4		Vía aérea	431.3
27	Vía terrestre	133.7		Vía terrestre	646.1
28	Turistas fronterizos	1,038.1		Turistas fronterizos	473.4
29	Peatones	192.3		Peatones	87.8
30	En automóviles	845.7		En automóviles	385.6
31					
32	Excursionistas internacionales	4,462.5		Excursionistas internacionales	6,256.2
33	Excursionistas fronterizos	4,205.9		Excursionistas fronterizos	6,256.2
34	Peatones	876.3		Peatones	1,517.9
35	En automóviles	3,329.7		En automóviles	4,738.3
36					
37	GASTO MEDIO (DOLARES)	157.6		GASTO MEDIO (DOLARES)	107.7
38					
39	Turistas internacionales	413.3		Turistas internacionales	372.6
40	Turistas de internación	769.8		Turistas de internación	513.1
41	Vía aérea	837.5		Vía aérea	968.5
42	Vía terrestre	310.9		Vía terrestre	209.0
43	Turistas fronterizos	55.8		Turistas fronterizos	52.9
44	Peatones	53.6		Peatones	32.3
45	En automóviles	56.3		En automóviles	57.6
46					
47	Excursionistas internacionales	38.4		Excursionistas internacionales	42.1
48	Excursionistas fronterizos	35.8		Excursionistas fronterizos	42.1
49	Peatones	33.9		Peatones	37.5
50	En automóviles	36.3		En automóviles	43.6
..					

Figura 1.3: Ejemplar Fuente 2 de Información Externa

NUMERO DE VIAJEROS (MILES)	6,542	S6 = S3 + S5
Turistas internacionales	2,079	S3 = S1 + S2
Turistas de internación	1,041	S1 = Dato1 + Dato2
Vía aérea	907	Dato1
Vía terrestre	134	Dato2
Turistas fronterizos	1,038	S2 = Dato3 + Dato4
Peatones	192	Dato3
En automóviles	846	Dato4
Excursionistas internacionales	4,462	S5 = S4 + Dato7
Excursionistas fronterizos	4,206	S4 = Dato5 + Dato6
Peatones	876	Dato5
En automóviles	3,330	Dato6

Figura 1.4: Ejemplar Fuente 2 de Información Externa, especificación

- En los bloques, la primera columna es para el nombre del organismo y las siguientes dos son datos acerca de sus actividades.
- El bloque de arriba tiene diecinueve filas y tres columnas.
 - * Contiene registros de dieciocho organismos y el dato de sus actividades.
 - * Aparece una fila en negritas con el total de los registros.
- El bloque de abajo tiene doce filas y tres columnas.
 - * Contiene registros de once organismos y el dato de sus actividades.
 - * Aparece una fila en negritas con el total de los registros.
- Para indicar que no hay valores en las actividades, se pone un “*” al principio del nombre y un “-“ en la parte de los datos.
- Al principio del documento hay una fila que contiene el total de totales por cada columna.

En la Figura 1.5 se puede ver la imagen del archivo.

Fuente 4

Son datos acerca de las visitas nacionales y extranjeras a establecimientos culturales y de recreación en México, vienen en un archivo de hoja de cálculo .

	A	B	C
1	ago-13		
2			
3			
4	PUERTO	ARRIBOS	PASAJEROS
5			
6	TOTAL	74	246,629
7			
8	PACIFICO	12	25,759
9			
10	*Rosarito, B.C.	-	-
11	Ensenada, B.C.	9	23,876
12	*San Carlos, B.C.S.	-	-
13	*Cabo San Lucas, B.C.S.	-	-
14	*Pichilingue, B.C.S.	-	-
15	*La Paz, B.C.S.	-	-
16	*Puerto Escondido, B.C.S.	-	-
17	*Loreto, B.C.S.	-	-
18	*Santa Rosalía, B.C.S.	-	-
19	*Guaymas, Son.	-	-
20	*Topolobampo, Sin.	-	-
21	*Mazatlán, Sin.	-	-
22	Vallarta, Jal.	1	627
23	*Manzanillo, Col.	-	-
24	*Zihuatanejo, Gro.	-	-
25	*Acapulco, Gro.	-	-
26	Huatulco, Oax.	1	628
27	Puerto Chiapas, Chis.	1	628
28			
29	GOLFO - CARIBE	62	220,870
30			
31	*Altamira, Tamps.	-	-
32	*Veracruz, Ver.	-	-
33	*Dos Bocas, Tab.	-	-
34	*Seybaplaya, Camp.	-	-
35	Progreso, Yuc.	8	23,717
36	*Cancun, Q.Roo	-	-
37	*Morelos, Q.Roo	-	-
38	*Playa del Carmen, Q. Roo	-	-
39	Cozumel, Q.Roo	50	180,214
40	*Punta Venado, Q.Roo	-	-
41	Majahual, Q.Roo	4	16,939

Figura 1.5: Ejemplar Fuente 3 de Información Externa

- El archivo consta de una sola pestaña.
- Hay una celda título (hasta arriba, cuarta columna) que indica el mes y el año (período).
- Hay datos para un número variable de filas y cinco columnas:
 - La primera indica un campo de localización geográfica.
 - La segunda indica un tipo de organismo.
 - La tercera indica el nombre del organismo.
 - La cuarta y quinta indican datos acerca de las actividades de los organismos para ese período.

En la Figura 1.6 se puede ver la imagen del archivo.

Fuente 5

Son datos acerca de vuelos y asientos programados desde aeropuertos nacionales e internacionales hacia México, vienen en archivo de bases de datos en Microsoft Access.

- Hay datos para un número variable de filas y veintisiete columnas:
 - Las primeras dos columnas indican año y mes (período).
 - Las siguientes veinticinco columnas indican, entre otras cosas, localizaciones geográficas, nombres de organismos así como sus atributos y actividades para ese período.
- Es un gran número de filas (aproximadamente 30,000).

En la Figura 1.7 se puede ver la imagen del archivo.

	A	C	D	E	F
1				feb-13	
2	Estado	Tipo	Centro	NAC.	EXT.
3	Baja California	Z.A.	Z.A. de El Vallecito	828	0
4	Baja California Sur	Z.A.	Z.A. de San Francisco de la	148	100
5	Campeche	Z.A.	Z.A. de Balamkú	224	188
6	Campeche	Z.A.	Z.A. de Becán	431	1,118
7	Campeche	Z.A.	Z.A. de Calakmul	555	1,245
8	Campeche	Z.A.	Z.A. de Chicanná	172	448
9	Campeche	Z.A.	Z.A. de Chunhuhub	72	4
10	Campeche	Z.A.	Z.A. de Dzibilnocac	149	8
11	Campeche	Z.A.	Z.A. de Edzná	2,345	761
12	Campeche	Z.A.	Z.A. de El Tigre	625	4
13	Campeche	Z.A.	Z.A. de Hochob	96	32
14	Campeche	Z.A.	Z.A. de Hormiguero	109	42
15	Campeche	Z.A.	Z.A. de Kankí	15	0
16	Campeche	Z.A.	Z.A. de Madzcaan	0	0
17	Campeche	Z.A.	Z.A. de Santa Rosa Xtampa	96	9
18	Campeche	Z.A.	Z.A. de Tohcok	60	33
19	Campeche	Z.A.	Z.A. de Xcalumkín	180	5
20	Campeche	Z.A.	Z.A. de Xpuhil	343	305
21	Campeche	Z.A.	Z.A. de Tabasqueño	171	17
22	Chiapas	Z.A.	Z.A. de Bonampak	3,874	615
23	Chiapas	Z.A.	Z.A. de Chincultic	1,344	176
24	Chiapas	Z.A.	Z.A. de Izapa	1,238	240
25	Chiapas	Z.A.	Z.A. de Palenque con museo	40,150	8,969
26	Chiapas	Z.A.	Z.A. de Tenam Fuente	997	48
27	Chiapas	Z.A.	Z.A. de Toniná con museo c	2,731	73
28	Chiapas	Z.A.	Z.A. de Yaxchilán	2,727	189
29	Chiapas	Z.A.	Z.A. de Chiapa de Corzo	369	14
30	Chiapas	Z.A.	Z.A de Laqartero	1,409	97

Figura 1.6: Ejemplar Fuente 4 de Información Externa

AÑO	N	TIPO_SERV	SER	VUELO	AE	NUM	DE	DEP_AIRP	DE	OME	REGIO	Pais	AR	ARR_Ali	AF	ARR_I	ESC	RUTA	FI	ASI	
2014	03	Doméstico	DXX	402701	40	Int 2701	MEX	México DF	MX	M SIN OME	México	México DF	ACA	Acapulco	MX	Mexico	00	Direct	MEXACA	13	1950
2014	05	Doméstico	DXX	402701	40	Int 2701	MEX	México DF	MX	M SIN OME	México	México DF	ACA	Acapulco	MX	Mexico	00	Direct	MEXACA	12	1800
2014	08	Doméstico	DXX	402701	40	Int 2701	MEX	México DF	MX	M SIN OME	México	México DF	ACA	Acapulco	MX	Mexico	00	Direct	MEXACA	13	1950
2014	11	Doméstico	DXX	402701	40	Int 2701	MEX	México DF	MX	M SIN OME	México	México DF	ACA	Acapulco	MX	Mexico	00	Direct	MEXACA	13	1950
2014	01	Doméstico	DXX	402701	40	Int 2701	MEX	México DF	MX	M SIN OME	México	México DF	ACA	Acapulco	MX	Mexico	00	Direct	MEXACA	13	1950
2014	04	Doméstico	DXX	402701	40	Int 2701	MEX	México DF	MX	M SIN OME	México	México DF	ACA	Acapulco	MX	Mexico	00	Direct	MEXACA	14	2100
2014	07	Doméstico	DXX	402701	40	Int 2701	MEX	México DF	MX	M SIN OME	México	México DF	ACA	Acapulco	MX	Mexico	00	Direct	MEXACA	14	2100
2014	10	Doméstico	DXX	402701	40	Int 2701	MEX	México DF	MX	M SIN OME	México	México DF	ACA	Acapulco	MX	Mexico	00	Direct	MEXACA	13	1950
2014	03	Doméstico	DXX	402703	40	Int 2703	MEX	México DF	MX	M SIN OME	México	México DF	ACA	Acapulco	MX	Mexico	00	Direct	MEXACA	23	3450
2014	06	Doméstico	DXX	402703	40	Int 2703	MEX	México DF	MX	M SIN OME	México	México DF	ACA	Acapulco	MX	Mexico	00	Direct	MEXACA	22	3300
2014	09	Doméstico	DXX	402703	40	Int 2703	MEX	México DF	MX	M SIN OME	México	México DF	ACA	Acapulco	MX	Mexico	00	Direct	MEXACA	21	3150
2014	06	Doméstico	DXX	402701	40	Int 2701	MEX	México DF	MX	M SIN OME	México	México DF	ACA	Acapulco	MX	Mexico	00	Direct	MEXACA	13	1950
2014	12	Doméstico	DXX	402703	40	Int 2703	MEX	México DF	MX	M SIN OME	México	México DF	ACA	Acapulco	MX	Mexico	00	Direct	MEXACA	21	3150
2014	02	Doméstico	DXX	402703	40	Int 2703	MEX	México DF	MX	M SIN OME	México	México DF	ACA	Acapulco	MX	Mexico	00	Direct	MEXACA	20	3000
2014	05	Doméstico	DXX	402703	40	Int 2703	MEX	México DF	MX	M SIN OME	México	México DF	ACA	Acapulco	MX	Mexico	00	Direct	MEXACA	23	3450
2014	09	Doméstico	DXX	402701	40	Int 2701	MEX	México DF	MX	M SIN OME	México	México DF	ACA	Acapulco	MX	Mexico	00	Direct	MEXACA	13	1950
2014	12	Doméstico	DXX	402701	40	Int 2701	MEX	México DF	MX	M SIN OME	México	México DF	ACA	Acapulco	MX	Mexico	00	Direct	MEXACA	14	2100
2014	01	Doméstico	DXX	402703	40	Int 2703	MEX	México DF	MX	M SIN OME	México	México DF	ACA	Acapulco	MX	Mexico	00	Direct	MEXACA	22	3300
2014	04	Doméstico	DXX	402703	40	Int 2703	MEX	México DF	MX	M SIN OME	México	México DF	ACA	Acapulco	MX	Mexico	00	Direct	MEXACA	20	3000
2014	03	Doméstico	DXX	40701	40	Int 701	TLC	Toluca	MX	M SIN OME	México	Toluca	ACA	Acapulco	MX	Mexico	00	Direct	TLCACA	23	3450
2014	06	Doméstico	DXX	40701	40	Int 701	TLC	Toluca	MX	M SIN OME	México	Toluca	ACA	Acapulco	MX	Mexico	00	Direct	TLCACA	22	3300
2014	01	Doméstico	DXX	40701	40	Int 701	TLC	Toluca	MX	M SIN OME	México	Toluca	ACA	Acapulco	MX	Mexico	00	Direct	TLCACA	22	3300
2014	04	Doméstico	DXX	40701	40	Int 701	TLC	Toluca	MX	M SIN OME	México	Toluca	ACA	Acapulco	MX	Mexico	00	Direct	TLCACA	20	3000
2014	07	Doméstico	DXX	40701	40	Int 701	TLC	Toluca	MX	M SIN OME	México	Toluca	ACA	Acapulco	MX	Mexico	00	Direct	TLCACA	21	3150
2014	10	Doméstico	DXX	40701	40	Int 701	TLC	Toluca	MX	M SIN OME	México	Toluca	ACA	Acapulco	MX	Mexico	00	Direct	TLCACA	22	3300
2014	05	Doméstico	DXX	40701	40	Int 701	TLC	Toluca	MX	M SIN OME	México	Toluca	ACA	Acapulco	MX	Mexico	00	Direct	TLCACA	23	3450
2014	08	Doméstico	DXX	40701	40	Int 701	TLC	Toluca	MX	M SIN OME	México	Toluca	ACA	Acapulco	MX	Mexico	00	Direct	TLCACA	23	3450
2014	11	Doméstico	DXX	40701	40	Int 701	TLC	Toluca	MX	M SIN OME	México	Toluca	ACA	Acapulco	MX	Mexico	00	Direct	TLCACA	22	3300

Figura 1.7: Ejemplar Fuente 5 de Información Externa

Fuente 6

Son datos acerca de llegadas de vuelos a México por nacionalidad, vienen en archivo de bases de datos en Microsoft Access.

- Hay datos para un número variable de filas y seis columnas:
 - Las primeras dos columnas indican año y mes (período).
 - La columna 3 indica un campo de identifica al aeropuerto.
 - Las columnas 4 y 5 indican un segundo campo (un campo compuesto) de localización geográfica.
 - La columna 6 indica datos acerca de las actividades de los organismos para ese período.
- Es un gran número de filas (aproximadamente 30,000).

En la Figura 1.8 se puede ver la imagen del archivo.

AÑO	MES	HUB	REGION	PAIS	ENTRADAS
2012	1	Chihuahua	Europa Occide	Noruega	1
2012	1	Chihuahua	Europa Occide	Portugal	1
2012	1	Chihuahua	Europa Occide	Reino Unido	34
2012	1	Chihuahua	Europa Occide	Suecia	1
2012	1	Chihuahua	Europa Occide	Suiza	2
2012	1	Chihuahua	Oceanía	Australia	4
2012	1	Chihuahua	Oceanía	Nueva Zelande	1
2012	1	Chihuahua	Sudamérica	Argentina	12
2012	1	Chihuahua	Sudamérica	Brasil	3
2012	1	Chihuahua	Sudamérica	Chile	2
2012	1	Chihuahua	Sudamérica	Colombia	1
2012	1	Chihuahua	Sudamérica	Perú	10
2012	1	Chihuahua	Sudamérica	Venezuela	4
2012	1	Ciudad del Carmen	África	Egipto	3
2012	1	Ciudad del Carmen	América del N	Canadá	22
2012	1	Ciudad del Carmen	América del N	Estados Unido:	818
2012	1	Ciudad del Carmen	Asia	China	3
2012	1	Ciudad del Carmen	Asia	Filipinas	2
2012	1	Ciudad del Carmen	Asia	India	6
2012	1	Ciudad del Carmen	Centro Améric	Cuba	1
2012	1	Ciudad del Carmen	Centro Améric	Honduras	1
2012	1	Ciudad del Carmen	Centro Améric	Nicaragua	1

Figura 1.8: Ejemplar Fuente 6 de Información Externa

1.2.2 Captura de Información Básica

Se debía poder capturar datos vía web para aquellos organismos de alojamiento que el cliente haya registrado previamente en su base de datos; a dichos organismos se les debía otorgar una clave y contraseña para poder acceder al sistema y hacer su respectiva captura de datos.

De acuerdo al cliente, se debía poder analizar y evaluar el comportamiento de la utilización de los servicios turísticos de hospedaje por medio de las siguientes variables:

- **Cuartos registrados** Número de habitaciones o cuartos con los que cuenta en su totalidad el establecimiento(organismo de servicios de alojamiento).
- **Cuartos disponibles** Número de habitaciones o cuartos con los que opera al día el establecimiento(organismo de servicios de alojamiento).
- **Cuartos ocupados por residentes** Número de habitaciones o cuartos ocupados o rentados durante el día por turistas que residen en el país.
- **Cuartos ocupados por no residentes** Número de habitaciones o cuartos ocupados o rentados durante el día por turistas que no residen en el país.
- **Llegada de turistas residentes** Número de turistas residentes que llegaron y ocuparon los cuartos ocupados residentes durante el día.
- **Llegada de turistas no residentes** Número de turistas no residentes que llegaron y ocuparon los cuartos ocupados no residentes durante el día.
- **Turistas noche de residentes** Es la suma de las llegadas de turistas residentes del día, más los turistas residentes que permanecen en el establecimiento del día anterior.
- **Turistas noche de no residentes** Es la suma de las llegadas de turistas no residentes del día, más los turistas no residentes que permanecen en el establecimiento del día anterior.

El usuario debía poder capturar cada uno de esos datos, teniendo en cuenta las siguientes reglas de negocio:

- El número de cuartos disponibles nunca debe ser mayor al número de cuartos registrados.

- El número de cuartos ocupados residentes siempre debe ser mayor o igual al número de turistas noche residentes.
- El número de cuartos ocupados no residentes siempre debe ser mayor o igual al número de turistas noche no residentes.
- La suma de cuartos ocupados residentes más los cuartos ocupados no residentes debe ser menor o igual al número de disponibles.
- El número de llegadas de turistas residentes siempre deber ser menor o igual número de turistas noche residentes.
- El número de llegadas de turistas no residentes siempre deber ser menor o igual número de turistas noche no residentes.

El sistema debía poder guardar estos datos por cada día del año y para cada organismo. Los usuarios debían poder capturar sus datos diariamente; además cada semana y cada mes se hacía un corte acerca de lo que el sistema tenía capturado de todos los organismos hasta ese momento. Los cortes debían ser registrados, así como la fecha en que se realizó, el grupo de organismos involucrado, la semana o mes que abarcó el corte y el usuario que realizó el corte.

Debido a que se trataba de una captura muy recurrente y a que los usuarios podían no estar familiarizados a cuestiones tecnológicas (usuarios básicos o principiantes), el sistema debía guiarlos por medio de una interfaz amigable que les facilitara su tarea. Se debían tener un sistema de validación al vuelo que indicara de manera gráfica en caso que la captura haya incumplido alguna de las reglas.

Por otra parte, debido a que no todos los usuarios podían capturar datos de los organismos, se debía poder aplicar una fórmula de expansión para completar datos faltantes con la intención de dar una aproximación de resultados a la que se tuviese con la participación todos los organismos.

1.2.3 Reportes

Los datos capturados debían ser procesados y posteriormente explotados, de tal manera que fuesen de utilidad para el cliente. A grandes rasgos, el tipo de reportes era sobre las diferentes actividades que realizaban los organismos durante cierto período y su comportamiento a lo largo del tiempo, de tal manera que fuese posible extraer información novedosa y útil acerca de sus datos. Aquí algunos ejemplos:

- Flujo de vuelos en actividad aeroportuaria.
- Llegada de visitantes por nacionalidad.
- Movimiento de cruceros en los principales puertos del país.
- Oferta de servicios de alojamiento por entidad federativa.
- Visitantes a zonas arqueológicas y museos.
- Monitoreo de actividad hotelera en los principales destinos turísticos del país.

En el Apéndice D se pueden ver imágenes de los reportes, producto del funcionamiento del sistema.

En la Figura 1.9 se presenta un esquema del entorno del sistema a desarrollar. Organismos y personal del Área de Información Básica debían poder enviar sus datos por medio de captura web. En el Área de Información Externa se obtenían los datos de los organismos, por medio de archivos (.xls y .mdb), y posteriormente se debían poder cargar al sistema. Luego de la obtención de datos, se debía procesar en el sistema la extracción de información valiosa para la toma de decisiones.

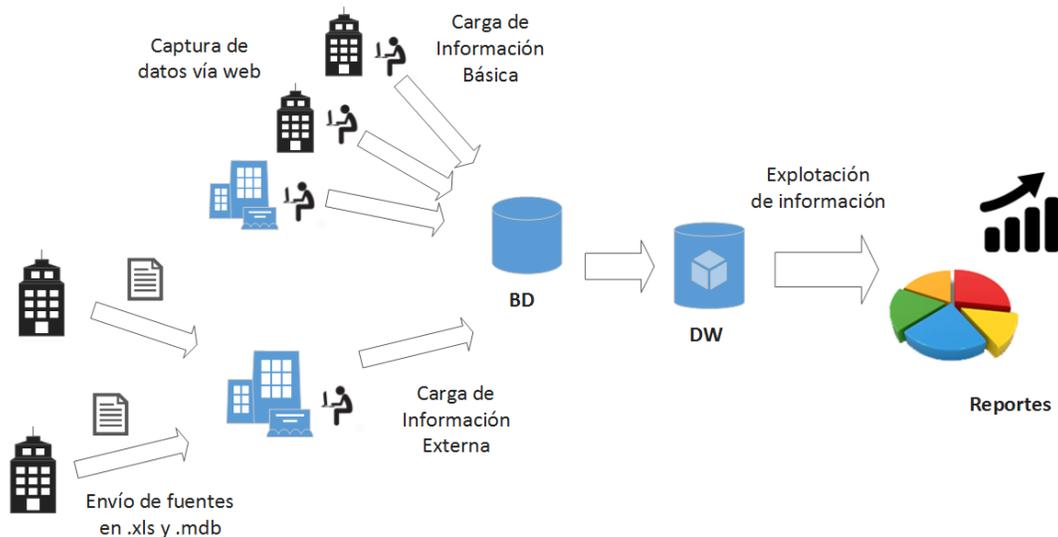


Figura 1.9: Entorno del cliente

1.2.4 Administración de Catálogos

Este módulo debía manejar los catálogos de aquellas entidades relacionadas al modelo del negocio. Entre éstas se encuentran:

- Localización geográfica (Países, Estados, Municipios, Zonas, Regiones).
- Organismos de servicios de alojamiento, las actividades que desarrollan, clasificación y agrupamientos.
- Organismos de recreación y culturales, así como sus tipos.
- Organismos de servicios aeroportuarios, aerolíneas, tipos de servicio, tipos de operación.
- Organismos puertos.
- Indicadores de evaluación.

No se guardaban datos de los organismos de servicios de administración migratoria y ni de control monetario.

Los catálogos debían permitir las opciones de alta, baja², modificación y consulta de datos. Se pueden ver los ejemplos de las operaciones de un catálogo en las Figuras del Apéndice C.

1.2.5 Administración del Sistema

Mediante niveles de seguridad el sistema debía permitir el correcto acceso a funciones y módulos destinados para su operación; así como tener controles que auditasen su utilización. Debía atender a una gran cantidad de usuarios, de los cuales se distinguen los siguientes tipos:

- Usuarios locales
Debían capturar datos de los organismos de servicios de alojamiento.
- Usuarios municipales
Debían supervisar y validar datos de los usuarios locales.

²Las bajas solamente se debían realizar de manera lógica, pues ningún dato ingresado en el sistema debía ser eliminado completamente.

- Usuarios estatales
Debían supervisar y validar, a su vez, datos de los usuarios municipales.
- Usuarios administradores
Debían capturar, supervisar y validar los datos de cualquier tipo de organismo. Se encargarían de generar reportes y tener un completo control sobre el sistema.

Los usuarios locales, municipales y estatales pertenecen al área de Información Básica. Los usuarios administradores pueden ser de ambas áreas.

1.3 Requerimientos no funcionales

El sistema a desarrollar debe correr sobre el sistema operativo Windows Server 2012 y apoyarse del sistema administrador de base de datos Microsoft SQL Server 2012.

Además el sistema debe poder visualizarse por lo menos en el navegador Internet Explorer versión 8, sin importar el sistema operativo de la computadora del usuario.

1.4 Antecedentes del sistema

El cliente llevaba a cabo sus tareas de reunir, procesar y difundir datos acerca de las actividades económicas vinculadas al turismo en México con ayuda de un sistema predecesor, el cual estaba desarrollado con el lenguaje de programación Java y el sistema manejador de base de datos Oracle.

Con este sistema, el cliente podía administrar sus catálogos, administrar permisos de acceso, cargar archivos externos con formato .csv, capturar datos vía página web y configurar *widgets* para visualizar reportes. Sin embargo, este software no satisfacía del todo sus necesidades; un análisis previo nos permitió identificar que tenía varios componentes sin funcionar y algunos de ellos, con un mal diseño, hacían un sistema demasiado lento e impredecible en cuanto a errores. Por lo cual, usuarios no-expertos debían realizar tareas de forma manual y sumamente peligrosas ya que ponían en riesgo la integridad de los datos; se realizaban tareas, tales como: consultas y administración de datos por medio de *scripts* en SQL, así como, limpieza y transformación de datos de archivos de hoja de cálculo y de base de datos (.xls y .mdb) a archivos separados por comas (.csv), para luego, cargarlos

en el sistema.

Cabe mencionar que el sistema predecesor no presentaba documentación en los códigos fuente del sistema y que, el cliente describía lo que pretendía del nuevo sistema con mucho detalle. Es por eso que, nuestro equipo de desarrollo se enfocó en homologar los resultados que otorgaban los módulos que sí funcionaban en el anterior, eliminar componentes obsoletos y agregar nuevos que facilitasen las tareas de los usuarios.

Dichos componentes se presentan en el siguiente capítulo.

Capítulo 2

Diseño e implementación del sistema

En este capítulo se especifican los detalles del diseño e implementación del sistema, como los son la plataforma de desarrollo y los componentes del sistema. Se usan diagramas de casos de uso, de bases de datos, del data warehouse, de la arquitectura del sistema y se presentan los algoritmos para los procesos de captura de datos del área de Información Básica y Externa.

2.1 Diseño del Sistema

A continuación se describen los módulos que presentó el nuevo sistema y que atendieron a los requerimientos descritos en el capítulo anterior.

Las siglas ABMC indican alta, baja, modificación y consulta.

- **Administración del sistema**

Se refiere a los componentes que permiten la correcta utilización y acceso a las funcionalidades del sistema.

- **ABMC Usuarios**

El catálogo de usuarios permite las operaciones de alta, baja, modificación y consulta. Se almacenan datos tales como: nombre, apellido paterno y materno, correo electrónico, rol, login y contraseña. Importante: un usuario administrador no puede modificar, editar ni consultar a otro usuario con rol administrador.

- **ABMC Roles**

El catálogo de roles permite las operaciones de alta, baja, modificación y

consulta. Se almacenan datos tales como: nombre del rol y descripción. De manera predeterminada, se tienen cargados los roles: rol usuario local, rol usuario municipal, rol usuario estatal y rol usuario administrador.

– **ABMC Módulos**

El catálogo de módulos permite las operaciones de alta, baja, modificación y consulta. Se refiere a cada uno de los módulos del sistema, por ejemplo los mismos catálogos. Se almacenan datos tales como: nombre del módulo y descripción.

– **Asignación de permisos**

Esta funcionalidad permite asociar los permisos de acceso de cada uno de los módulos con los roles del sistema.

– **ABMC Mensajes**

El catálogo de mensajes permite las operaciones de alta, baja, modificación y consulta. Permite mandar mensajes de aviso en un determinado período de tiempo. Los mensajes sólo los manda un usuario de rol administrador hacia usuarios estatales, municipales o locales. Se almacenan datos tales como: rol de usuario destinatario, mensaje, fecha de inicio y fecha fin.

• **Catálogos generales**

Estos catálogos permiten restringir y agrupar los datos que se tienen en el sistema.

– **ABMC Países**

El catálogo de países permite las operaciones de alta, baja, modificación y consulta. Se almacenan datos tales como: nombre del país y código.

– **ABMC Entidades federativas**

El catálogo de entidades federativas permite las operaciones de alta, baja, modificación y consulta. Se almacenan datos tales como: nombre de la entidad y país al que pertenece.

– **ABMC Municipios**

El catálogo de municipios permite las operaciones de alta, modificación, baja y consulta. Se almacenan datos tales como: nombre del municipio, entidad federativa y país al que pertenece.

– **ABMC Zonas**

El catálogo de zonas permite las operaciones de alta, baja, modificación y consulta. Se almacena sólo el nombre de la zona. Algunos ejemplos de

zonas son: América del Norte, América Central, Europa, Este de Asia y Pacífico, África.

– **ABMC Regiones**

El catálogo de regiones permite las operaciones de alta, baja, modificación y consulta. Se almacenan datos tales como: nombre de la región y zona a la que pertenece. Las regiones se pueden referir a ciudades, estados o países; por ejemplo: Vaticano, New York, Islas Fuji.

– **ABMC Actividades de organismos de alojamiento**

El catálogo de actividades de organismos permite las operaciones de alta, baja, modificación y consulta. Se almacenan datos tales como: nombre de la actividad y descripción. Ejemplos de este catálogo pueden ser hoteles, moteles, villas, cabañas.

– **ABMC Clasificación de organismos de alojamiento**

El catálogo de clasificación de organismos de alojamiento permite las operaciones de alta, baja, modificación y consulta. Se almacenan datos tales como: nombre de la clasificación y actividad a la que pertenece. Los organismos están asociados a diferentes clasificaciones, dependiendo de su actividad. Un ejemplo de clasificación es la de categorías de hotel: 1,2,3,4 y 5 estrellas.

– **ABMC Grupos de organismos de alojamiento**

El catálogo de grupos de organismos de alojamiento permite las operaciones de alta, baja, modificación y consulta. Se refiere a agrupaciones de organismos que comparten características en común. Se almacenan datos tales como: nombre del grupo y descripción.

– **ABMC Organismos de alojamiento**

El catálogo de organismos de alojamiento permite las operaciones de alta, baja, modificación y consulta. Entre otras características, guarda datos tales como: nombre del organismo, fecha de inicio y fin de actividades, nombre y cargo del responsable, teléfono 1 y 2, fax, correo electrónico, sitio web, grupo al que está asociado, clasificación, actividad que desarrolla, municipio, entidad federativa y país al que pertenece. Cada cambio realizado en el registro de un organismo es guardado históricamente de manera interna.

– **ABMC Tipos de organismos de recreación o culturales**

El catálogo de tipos de organismos de recreación o culturales permite las operaciones de alta, baja, modificación y consulta. Se almacena solo el

nombre del tipo. Ejemplos de este catálogo pueden ser zonas arqueológicas, museos.

– **ABMC Organismos de recreación y culturales**

El catálogo de organismos de recreación y culturales permite las operaciones de alta, baja, modificación y consulta. Entre otras características, guarda datos tales como: clave y nombre del organismo, entidad federativa y país al que pertenece, y tipo de organismo de recreación o cultural.

– **ABMC Aerolíneas**

El catálogo de aerolíneas permite las operaciones de alta, baja, modificación y consulta. Entre otras características, guarda datos tales como: clave y nombre de la aerolínea.

– **ABMC Tipos de Servicio**

El catálogo de tipos de servicio permite las operaciones de alta, baja, modificación y consulta. Se almacenan datos tales como: clave y nombre del tipo de servicio. Se refiere a tipos de servicio que ofrecen las aerolíneas.

– **ABMC Organismos aeroportuarios**

El catálogo de organismos aeroportuarios permite las operaciones de alta, baja, modificación y consulta. Entre otras características, guarda datos tales como: clave y nombre del organismo, municipio, entidad federativa y país al que pertenece.

– **ABMC Organismos aeroportuarios**

El catálogo de organismos aeroportuarios permite las operaciones de alta, baja, modificación y consulta. Entre otras características, guarda datos tales como: clave y nombre del organismo, municipio, entidad federativa y país al que pertenece.

– **ABMC Tipos de Operación**

El catálogo de tipos de operación permite las operaciones de alta, baja, modificación y consulta. Se refiere a tipos de operación que ofrecen los organismos aeroportuarios.

– **ABMC Organismos Puertos**

El catálogo de organismos puertos permite las operaciones de alta, baja, modificación y consulta. Entre otras características, guarda datos tales como: clave y nombre del organismo y región a la que pertenece.

– **ABMC Indicadores de evaluación**

El catálogo de indicadores de evaluación permite las operaciones de alta,

baja, modificación y consulta. Se almacenan datos tales como: nombre del indicador y descripción. Contribuye en los métodos de evaluación para la generación de reportes.

- **Captación de información**

Este módulo permite a los usuarios ingresar datos específicos y concretos acerca de los organismos.

- **Captura de Información Básica**

Los datos pueden registrarse por medio de un formulario web.

- **Carga de Información Externa**

Los datos pueden cargarse al sistema por medio de archivos en Excel (.xls, .xlsx) o en Access (.mdb).

Ambos componentes se verán a detalle en las siguientes secciones.

- **Monitoreo de organismos de alojamiento**

- **Asignación de permisos de monitoreo**

Otorga permisos a los usuarios para *monitorear* organismos.

- * Los usuarios locales están asociados a un solo organismo de alojamiento, del cual estarán capturando datos de manera permanente; la opción de monitoreo no se les asigna.
- * Los usuarios municipales y estatales pueden monitorear organismos de alojamiento, acorde su localización geográfica.
- * Los usuarios administrativos pueden monitorear la totalidad de los organismos de alojamiento, como parte de sus privilegios.

Monitorear se refiere al proceso que realizan los usuarios para asegurarse del cumplimiento de la captura de datos. Los usuarios municipales llevan un control de los datos que han capturado sus¹ usuarios locales, si no han llevado a cabo su captura se les recuerda². De igual forma, del usuario estatal al municipal y del usuario administrador al estatal.

- **Monitoreo de cierre semanal y mensual**

Los cierres se llevan a cabo a nivel municipal y estatal. Este módulo permite a los usuarios realizar cierres de monitoreo sobre los organismos de alojamiento a los que esté asociado.

¹Es decir los usuarios locales que están asociados a organismos que pertenecen a esa región geográfica.

²Se le comunica por medio de los números telefónico o correo electrónico ingresados en el sistema

- * Un cierre semanal indica la captura de datos que se llevaron a cabo los siete días de una determinada semana del año.
- * Un cierre mensual indica la captura de datos que se llevaron a cabo en los días que contiene determinado mes del año.

– **Relación de Cierres**

Se lleva a cabo una bitácora de los cierres realizados o por realizar. Se almacenan datos tales como:

- * Tipo de cierre (semanal o mensual).
- * Grupo de organismos de alojamiento.
- * Año.
- * Semana o mes, según sea el caso.
- * Usuario que realizó el cierre.
- * Fecha de solicitud.
- * Fecha de realización.
- * Estatus de la operación.

• **Reportes**

La información valiosa obtenida de la captura de datos puede verse reflejada en Dashboards u otro tipo de reportes, con opción de descarga en hojas de cálculo en Microsoft Excel o en PDF, según el caso. Ver Apéndices C y D.

- Reportes de Información Básica.
- Reportes de Información Externa.

A continuación se presenta la Figura 2.1 que indica el diagrama general de casos de uso. Indica las tareas del sistema y su interacción con los usuarios.

- La tarea *Administración del Sistema* abarca las tareas *ABMC Usuarios*, *ABMC Roles*, *ABMC Módulos*, *Asignación de permisos* y *ABMC Mensajes*.
- De igual manera, la tarea *Administración de Catálogos* abarca las tareas *ABMC Países*, *ABMC Entidades Federativas*, *ABMC Municipios*, *ABMC Zonas*, *ABMC Regiones*, *ABMC Actividades de organismos de alojamiento*, *ABMC Clasificación de organismos de alojamiento*, *ABMC Grupos de organismos de alojamiento*, *ABMC Organismos de alojamiento*, *ABMC Tipos de organismos de recreación o culturales*, *ABMC Organismos de recreación y culturales*, *ABMC Aerolíneas*, *ABMC Tipos de servicio*, *ABMC Organismos aeroportuarios*, *ABMC Tipos de Operación*, *ABMC Organismos puertos*, y *ABMC Indicadores de evaluación*.

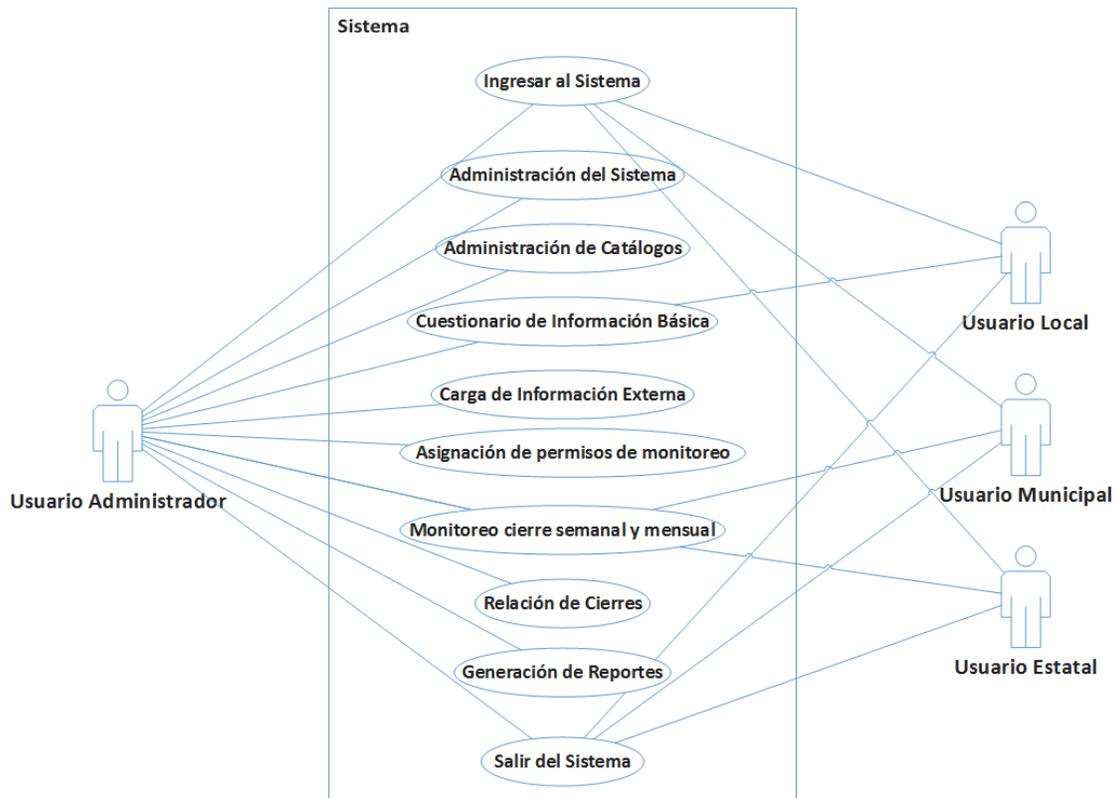


Figura 2.1: Diagrama general de casos de uso

Mi participación en el desarrollo del sistema, en concreto, fue llevar la responsabilidad de implementar todas las funcionalidades arriba mencionadas, excepto la sección de Reportes de Información Externa.

2.2 Diagrama de base de datos y de data warehouse

Tomando en cuenta que se requirió un sistema orientado a la recolección, almacenamiento, procesamiento y explotación de datos, además de un módulo de administración de los mismos, fue necesario dividir las operaciones del sistema: una para llevar a cabo la funcionalidad de soporte de decisiones, de la cual se hará

cargo el Procesamiento Analítico en Línea³, y otra para la operación diaria a cargo del Procesamiento de Transacciones En Línea.⁴

A continuación se presenta una tabla de apoyo:

Tabla 2.1: OLTP vs OLAP

Característica	OLTP	OLAP
Funcionalidad	Operación diaria	Soporte de decisiones
Diseño	Estructurado y orientado a la aplicación	Hecho a la medida y orientado a objetivos
Modelo	Relacional	Estrella, copo de nieve
Datos	Actualizados, aislados, no repetidos	Históricos y consolidados
Vista	Detallada	Resumida, múltiples dimensiones
Acceso	Lectura/Escritura	Sólo Lectura

En la Figura 2.2 y 2.3 podemos ver el diagrama de la base de datos operacional del sistema⁵. Este diagrama define las tablas, los campos que la componen y la relación entre tablas.

³OLAP, por sus siglas en inglés

⁴OLTP, por sus siglas en inglés

⁵La dividí en dos partes para que fuera visible en el documento. Se repiten las tablas Estados y Países, sólo como referencia en las relaciones.

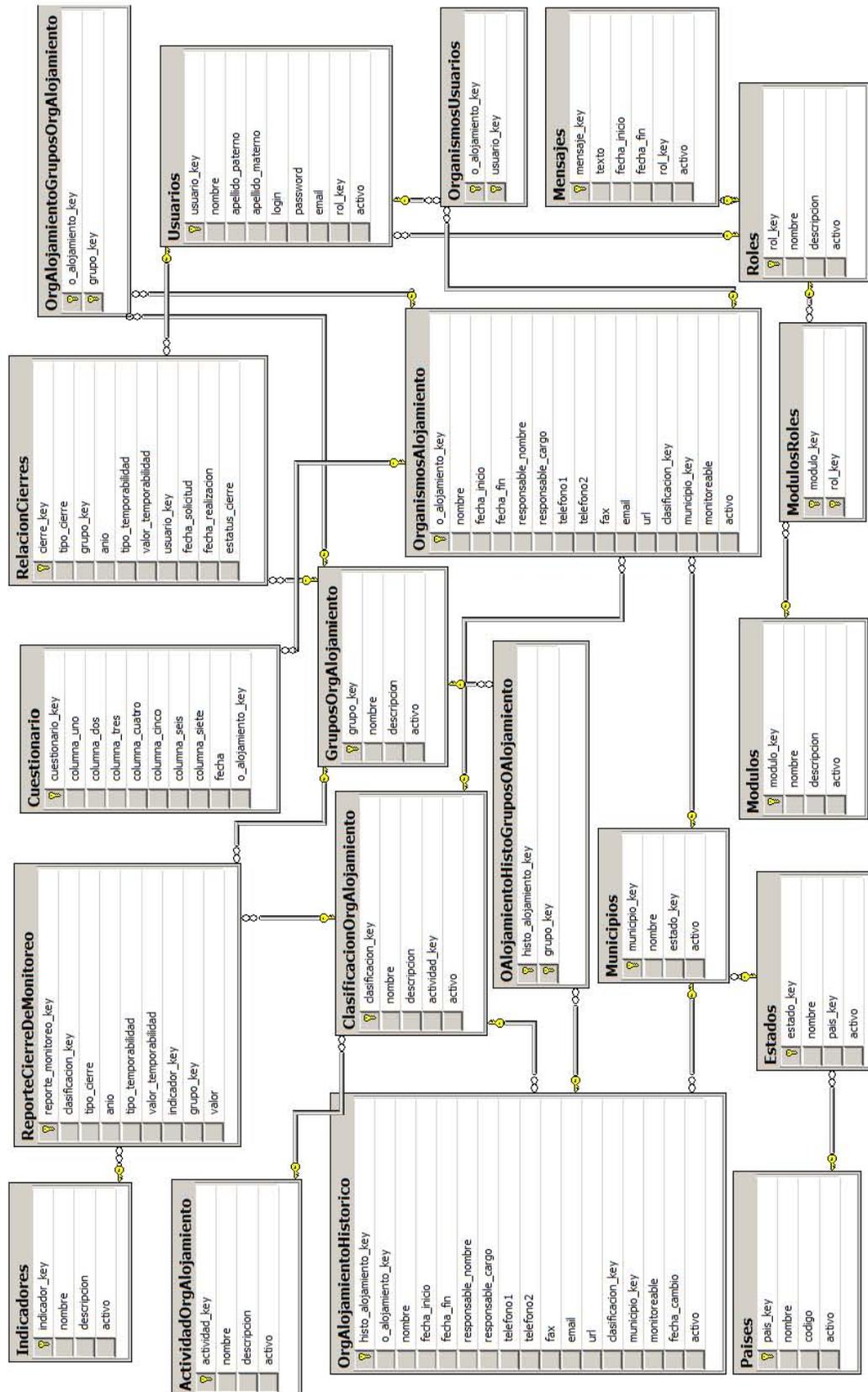
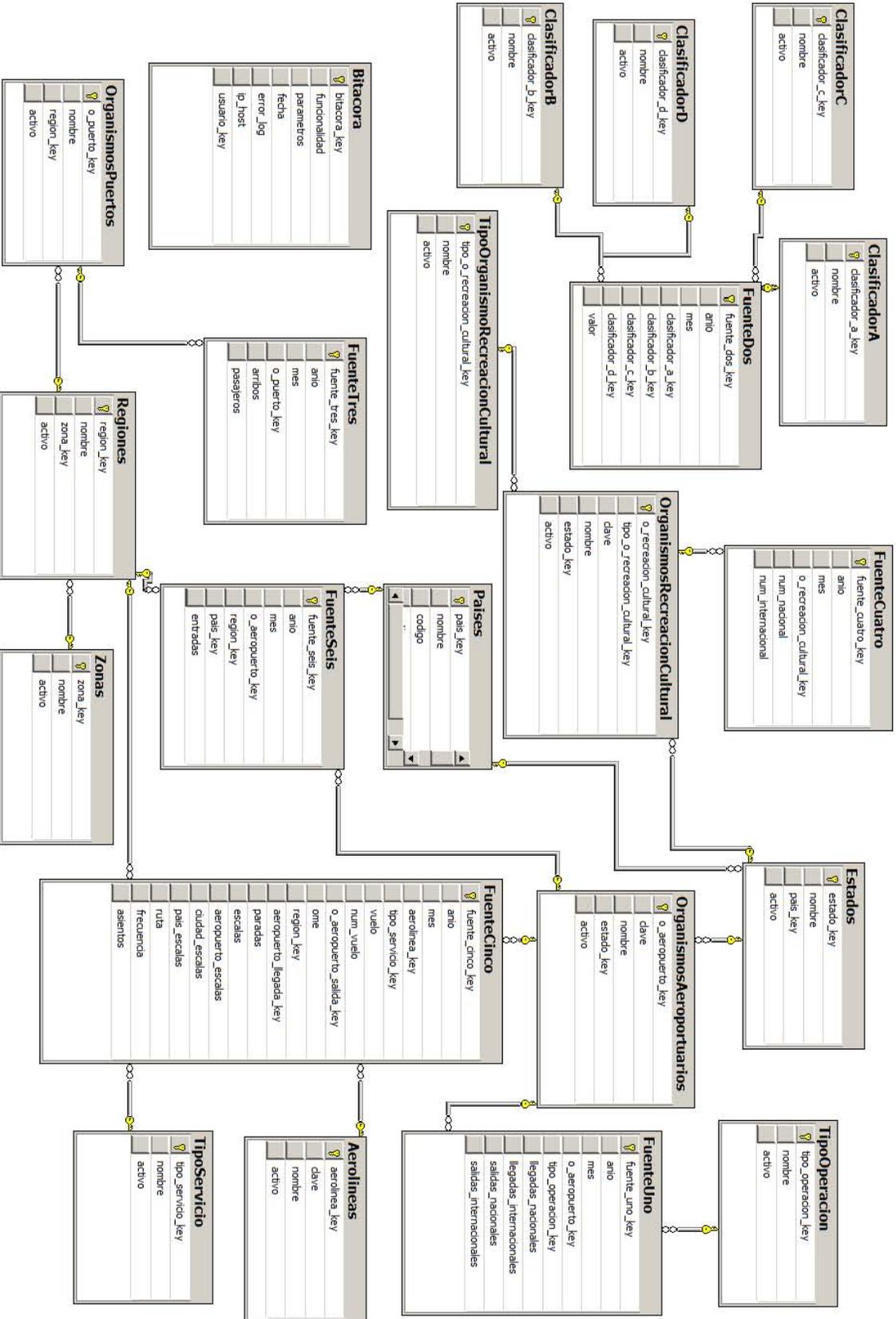


Figura 2.2: Diagrama de la base de datos, parte I

Figura 2.3: Diagrama de la base de datos, parte II



Se tienen las tablas:

1. **Aerolíneas**

Tabla catálogo para el componente *ABMC Aerolíneas*. Se guardan la *clave* (VARCHAR) y el *nombre* (VARCHAR) de la aerolínea. Además, se guarda la *llave primaria* (INT) de la tabla y un campo *activo* (BIT) que indica si el registro ha sido dado de baja o no.

2. **ActividadOrgAlojamiento**

Tabla catálogo para el componente *ABMC Actividades de organismos de alojamiento*. Se guardan el *nombre* (VARCHAR) y la *descripción* (VARCHAR) de la actividad. Además, se guarda la *llave primaria* (INT) de la tabla y un campo *activo* (BIT) que indica si el registro ha sido dado de baja o no.

3. **Bitacora**

Tabla de registro de incidencias. Se guardan la *funcionalidad* (VARCHAR), *parámetros* (VARCHAR), *fecha* (DATE) y *error log* (VARCHAR) de un posible error en el sistema que sucedió para una determinada *ip host* (VARCHAR) y *usuario del sistema* (INT). Además, se guarda la *llave primaria* (INT) de la tabla.

4. **ClasificadorA**

Tabla catálogo para la clasificación de datos que se extraen de los archivos en Excel de la *f fuente dos* de Información Externa. Se guarda el *nombre* (VARCHAR) del concepto, la *llave primaria* (INT) de la tabla y un campo *activo* (BIT) que indica si el registro ha sido dado de baja o no.

5. **ClasificadorB**

Tabla catálogo para la clasificación de datos que se extraen de los archivos en Excel de la *f fuente dos* de Información Externa. Se guarda el *nombre* (VARCHAR) del concepto, la *llave primaria* (INT) de la tabla y un campo *activo* (BIT) que indica si el registro ha sido dado de baja o no.

6. **ClasificadorC**

Tabla catálogo para la clasificación de datos que se extraen de los archivos en Excel de la *f fuente dos* de Información Externa. Se guarda el *nombre* (VARCHAR) del concepto, la *llave primaria* (INT) de la tabla y un campo *activo* (BIT) que indica si el registro ha sido dado de baja o no.

7. ClasificadorD

Tabla catálogo para la clasificación de datos que se extraen de los archivos en Excel de la *fuentes dos* de Información Externa. Se guarda el *nombre* (VARCHAR) del concepto, la *llave primaria* (INT) de la tabla y un campo *activo* (BIT) que indica si el registro ha sido dado de baja o no.

8. Cuestionario

Tabla de hechos donde se guardan 7 *columnas* con el tipo de dato (INT), con referencia a un *organismo de alojamiento* para determinada *fecha* (DATE). Los campos se verán a detalle más adelante.

9. ClasificacionOrgAlojamiento

Tabla catálogo para el componente *ABMC Clasificación de organismos de alojamiento*. Se guardan el *nombre* (VARCHAR), la *descripción* (VARCHAR) y la referencia a la *actividad* (INT) de la clasificación. Además, se guarda la *llave primaria* (INT) de la tabla y un campo *activo* (BIT) que indica si el registro ha sido dado de baja o no.

10. FuenteUno

Tabla de hechos donde se guardan los valores extraídos de la *fuentes uno* de Información Externa. Se guarda la *llave primaria* (INT) de la tabla, el campo *año* (INT) y *mes* (INT) del período que corresponde al archivo cargado; las referencias al *aeropuerto* (INT) en cuestión y su *tipo de operación*, así como, los valores del número de *llegadas nacionales* (INT) e *internacionales* (INT), *salidas nacionales* (INT) e *internacionales* (INT). Los campos se verán a detalle más adelante.

11. FuenteDos

Tabla de hechos donde se guardan los valores extraídos de la *fuentes dos* de Información Externa. Se guarda la *llave primaria* (INT) de la tabla, el campo *año* (INT) y *mes* (INT) del período que corresponde al archivo cargado; las referencias a cuatro clasificadores: al *clasificadorA* (INT), al *clasificadorB* (INT), al *clasificadorC* (INT), al *clasificadorD* (INT) y el *valor* (FLOAT) de la celda en cuestión. Los campos se verán a detalle más adelante.

12. FuenteTres

Tabla de hechos donde se guardan los valores extraídos de la *fuentes tres* de Información Externa. Se guarda la *llave primaria* (INT) de la tabla, el campo *año* (INT) y *mes* (INT) del período que corresponde al archivo cargado; las

referencias al *puerto* (INT) en cuestión, así como, los valores del número de *arribos* (INT) y de *pasajeros* (INT). Los campos se verán a detalle más adelante.

13. FuenteCuatro

Tabla de hechos donde se guardan los valores extraídos de la *fuentes cuatro* de Información Externa. Se guarda la *llave primaria* (INT) de la tabla, el campo *año* (INT) y *mes* (INT) del período que corresponde al archivo cargado; las referencias al *organismo de recreación o cultural* (INT) en cuestión, así como, los valores del número de visitas *nacionales* (INT) e *internacionales* (INT). Los campos se verán a detalle más adelante.

14. FuenteCinco

Tabla de hechos donde se guardan los valores extraídos de la *fuentes cinco* de Información Externa. Se guarda la *llave primaria* (INT) de la tabla, el campo *año* (INT) y *mes* (INT) del período que corresponde al archivo cargado; las referencias a la *aerolínea* (INT) y su *tipo de servicio* (INT), el *aeropuerto de salida* (INT), la *región* (INT) y el *aeropuerto de llegada* (INT) en cuestión, así como, el valor del *vuelo* (INT), el *número de vuelo* (INT), el valor *ome* (VARCHAR), el *número de paradas* (INT), *número de escalas* (INT), *los aeropuertos* (VARCHAR) y *ciudades* (VARCHAR) de escala y su *ruta* (VARCHAR), además del número de *asientos* (INT) y de *frecuencia* (INT). Los campos se verán a detalle más adelante.

15. FuenteSeis

Tabla de hechos donde se guardan los valores extraídos de la *fuentes seis* de Información Externa. Se guarda la *llave primaria* (INT) de la tabla, el campo *año* (INT) y *mes* (INT) del período que corresponde al archivo cargado; las referencias al *aeropuerto* (INT), la *región* (INT) y el *país* (INT) en cuestión, así como, el valor del número de *entradas* (INT). Los campos se verán a detalle más adelante.

16. GruposOrgAlojamiento

Tabla catálogo para el componente *ABMC Grupos de organismos de alojamiento*. Se guardan el *nombre* (VARCHAR) y la *descripción* (VARCHAR) del grupo. Además, se guarda la *llave primaria* (INT) de la tabla y un campo *activo* (BIT) que indica si el registro ha sido dado de baja o no.

17. Estados

Tabla catálogo para el componente *ABMC Estados*. Se guardan el *nombre*

(VARCHAR) y la referencia al *país* (INT) de la entidad federativa. Además, se guarda la *llave primaria* (INT) de la tabla y un campo *activo* (BIT) que indica si el registro ha sido dado de baja o no.

18. Indicadores

Tabla catálogo para el componente *ABMC Indicadores*. Se guardan el *nombre* (VARCHAR) y la *descripción* (VARCHAR) del indicador. Además, se guarda la *llave primaria* (INT) de la tabla y un campo *activo* (BIT) que indica si el registro ha sido dado de baja o no.

19. Mensajes

Tabla catálogo para el componente *ABMC Mensajes*. Se guardan el *texto* (VARCHAR) a mostrar, la *fecha inicio* y la *fecha fin* (DATE), así como el *usuario del sistema* (INT) para mostrarle el mensaje. Además, se guarda la *llave primaria* (INT) de la tabla y un campo *activo* (BIT) que indica si el registro ha sido dado de baja o no.

20. Modulos

Tabla catálogo para el componente *ABMC Módulos*. Se guardan el *nombre* (VARCHAR) y la *descripción* (VARCHAR) del módulo del sistema. Además, se guarda la *llave primaria* (INT) de la tabla y un campo *activo* (BIT) que indica si el registro ha sido dado de baja o no.

21. ModulosRoles

Tabla de asociación entre la tabla **Modulos** y la de **Roles**.

22. Municipios

Tabla catálogo para el componente *ABMC Municipios*. Se guardan el *nombre* (VARCHAR) y la referencia al *estado* (INT) del municipio. Además, se guarda la *llave primaria* (INT) de la tabla y un campo *activo* (BIT) que indica si el registro ha sido dado de baja o no.

23. OrganismosAeroportuarios

Tabla catálogo para el componente *ABMC Organismos aeroportuarios*. Se guardan la *clave* (VARCHAR), el *nombre* (VARCHAR) y la referencia al *estado* (INT) al que pertenece el organismo. Además, se guarda la *llave primaria* (INT) de la tabla y un campo *activo* (BIT) que indica si el registro ha sido dado de baja o no.

24. OrganismosAlojamiento

Tabla catálogo para el componente *ABMC Organismos de alojamiento*. Se

guardan el *nombre* (VARCHAR), la *fecha inicio* y la *fecha fin* (DATE) de actividades en el sistema, el *nombre* (VARCHAR) y *cargo del responsable* (VARCHAR), el *teléfono 1* (VARCHAR), *teléfono 2* (VARCHAR), *fax* (VARCHAR), *correo electrónico* (VARCHAR) y la *url de su página web* (VARCHAR), así como, las referencias a la *clasificación* (INT) y *municipio* (INT) al que pertenece el organismo. Además, se guarda la *llave primaria* (INT) de la tabla, una bandera *monitoreable* (BIT) que indica si el organismo es habilitado para ser monitoreado, y un campo *activo* (BIT) que indica si el registro ha sido dado de baja o no.

25. **OrganismosPuertos**

Tabla catálogo para el componente *ABMC Organismos puertos*. Se guardan el *nombre* (VARCHAR) y la referencia a la *region* (INT) al que pertenece el organismo. Además, se guarda la *llave primaria* (INT) de la tabla y un campo *activo* (BIT) que indica si el registro ha sido dado de baja o no.

26. **OrganismosRecreacionCultural**

Tabla catálogo para el componente *ABMC Organismos de recreación y culturales*. Se guardan la *clave* (VARCHAR), el *nombre* (VARCHAR), y las referencias al *tipo de organismo de recreación-cultural* (INT) y al *estado* (INT) al que pertenece el organismo. Además, se guarda la *llave primaria* (INT) de la tabla y un campo *activo* (BIT) que indica si el registro ha sido dado de baja o no.

27. **OrgAlojamientoGruposOrgAlojamiento**

Tabla de asociación entre la tabla **OrganismosAlojamiento** y la de **GruposOrAlojamiento**.

28. **OrgAlojamientoHistorico**

Tabla de históricos para el componente *ABMC Organismos*. Contiene los mismos campos que la tabla **OrganismosAlojamiento**, además de su propia *llave primaria* (INT) de tabla y un campo *fecha cambio* (DATE) que indica la fecha en que se aplicará la modificación.

29. **OAlojamientoHistoGruposOAlojamiento**

Tabla de asociación entre la tabla **OrgAlojamientoHistorico** y la de **GruposOrgAlojamiento**.

30. **OrganismosUsuarios**

Tabla de asociación entre la tabla **OrganismosAlojamiento** y la de **Usuarios**.

31. Países

Tabla catálogo para el componente *ABMC Países*. Se guardan el *código* (VARCHAR) y el *nombre* (VARCHAR) del país. Además, se guarda la *llave primaria* (INT) de la tabla y un campo *activo* (BIT) que indica si el registro ha sido dado de baja o no.

32. RelacionDeCierres

Tabla de hechos donde se guarda la bitácora de los cierres realizados o por realizar para el monitoreo de organismos de alojamiento. Se guarda la *llave primaria* (INT) de la tabla, el campo *tipo de cierre* (BIT), que indica si el cierre es preliminar o definitivo; el *año* (INT), una bandera *tipo de temporabilidad* (BIT) que indica si el *valor de temporabilidad* (INT) es un número que indica semana o mes; también se almacenan las referencias al *usuario del sistema* (INT) y al *grupo de organismos* (INT), así como las *fechas de solicitud y de realización* (DATE) del cierre.

33. Regiones

Tabla catálogo para el componente *ABMC Regiones*. Se guardan el *nombre* (VARCHAR) y la referencia a la *zona* (INT) de la región. Además, se guarda la *llave primaria* (INT) de la tabla y un campo *activo* (BIT) que indica si el registro ha sido dado de baja o no.

34. ReporteCierreDeMonitoreo

Tabla de hechos que guarda, de manera normalizada, los resultados que arroja un cierre de monitoreo de organismos de alojamiento. Estos datos se obtienen después de un proceso de muestreo que involucra tanto a los organismos de alojamiento que han capturado los resultados de sus actividades, por medio del módulo de Carga de Información Básica, como a los que no. Es decir, a partir de los datos capturados, por medio de una fórmula de expansión, se acompletan datos faltantes y se da una aproximación parecida a la que se tuviese con la participación todos los organismos. Se guarda la *llave primaria* (INT) de la tabla, el campo *tipo de cierre* (BIT), que indica si el cierre es preliminar o definitivo; el *año* (INT), una bandera *tipo de temporabilidad* (BIT) que indica si el *valor de temporabilidad* (INT) es un número que indica semana o mes; también se almacenan las referencias al *indicador* (INT) del reporte, el *grupo* (INT) y la *clasificación* (INT) de los organismos seleccionados, así como el *valor obtenido* (FLOAT) del proceso de expansión.

35. Roles

Tabla catálogo para el componente *ABMC Roles*. Se guardan el *nombre*

(VARCHAR) y la *descripción* (VARCHAR) del rol de usuario. Además, se guarda la *llave primaria* (INT) de la tabla y un campo *activo* (BIT) que indica si el registro ha sido dado de baja o no.

36. TipoOperacion

Tabla catálogo para el componente *ABMC Tipos de Operación*. Se guarda el *nombre* (VARCHAR) del tipo de operación de organismos aeroportuarios, la *llave primaria* (INT) de la tabla y un campo *activo* (BIT) que indica si el registro ha sido dado de baja o no.

37. TipoOrganismoRecreacionCultural

Tabla catálogo para el componente *ABMC Tipos de organismos de recreación o culturales*. Se guarda el *nombre* (VARCHAR) del tipo de organismo de recreación o cultural, así como, la *llave primaria* (INT) de la tabla y un campo *activo* (BIT) que indica si el registro ha sido dado de baja o no.

38. TipoServicio

Tabla catálogo para el componente *ABMC Tipos de Servicio*. Se guarda el *nombre* (VARCHAR) del tipo de servicio de aerolíneas, la *llave primaria* (INT) de la tabla y un campo *activo* (BIT) que indica si el registro ha sido dado de baja o no.

39. Usuarios

Tabla catálogo para el componente *ABMC Usuarios*. Se guardan el *nombre* (VARCHAR), *apellido paterno* (VARCHAR), *apellido materno* (VARCHAR), *login* (VARCHAR), *contraseña* (VARCHAR), *correo electrónico* (VARCHAR) y la referencia al *rol* (INT) de usuario. Además, se guarda la *llave primaria* (INT) de la tabla y un campo *activo* (BIT) que indica si el registro ha sido dado de baja o no.

40. Zonas

Tabla catálogo para el componente *ABMC Zonas*. Se guarda el *nombre* (VARCHAR) de la zona, la *llave primaria* (INT) de la tabla y un campo *activo* (BIT) que indica si el registro ha sido dado de baja o no.

El diagrama de la base de datos para el nuevo sistema fue el resultado del diagrama de la base de datos anterior y la adición de nuevas tablas y relaciones proporcionadas por nuestro equipo de desarrollo.

Nuestro equipo optó por respetar el diagrama de la base de datos anterior para

poder desarrollar los módulos de Administración de Catálogos y de Sistema así como el de Captura de Información Básica que ya presentaba el software predecesor.

Las principales diferencias entre la base de datos anterior y la actual fueron que, se añadieron las tablas vinculadas al funcionamiento del módulo de los procesos ETL. Es decir, se añadieron *Aerolineas*, *Bitacora*, *ClasificadorA*, *ClasificadorB*, *ClasificadorC*, *ClasificadorD*, *FuenteUno*, *FuenteDos*, *FuenteTres*, *FuenteCuatro*, *FuenteCinco*, *FuenteSeis*, *OrganismosAeroportuarios*, *OrganismosPuertos*, *OrganismosRecreacionCultural*, *Regiones*, *TipoOperacion*, *TipoServicio*, *Zonas* y *TipoOrganismoRecreacionCultural*.

En cuanto al diagrama del data warehouse, se utiliza el modelo copo de nieve para poder ahorrar espacio en tablas normalizadas. El diagrama se muestra en las siguientes siete figuras⁶.

En la Figura 2.4 podemos ver el diagrama del data warehouse que involucra los datos históricos de los organismos de alojamiento, así como los datos del cuestionario y del cierre de monitoreo.

⁶La dividí en partes para que fuera visible en el documento. Se repiten algunas tablas, sólo como referencia en las relaciones.

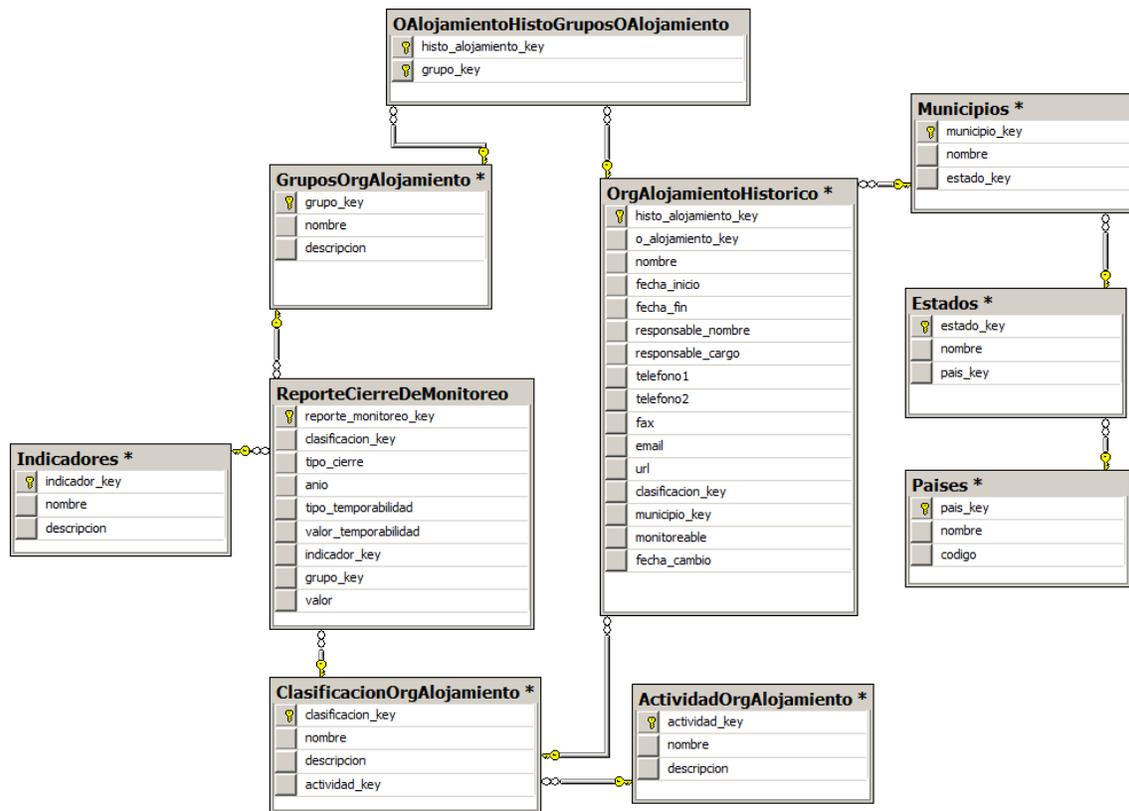


Figura 2.4: Diagrama del data warehouse, parte I

En la Figura 2.5 podemos ver el diagrama del data warehouse que involucra los datos de las fuentes 1 y 2 del área de Información Externa. En la Figura 2.6 podemos ver el diagrama del data warehouse que involucra los datos de las fuentes 3 y 4 del área de Información Externa. En la Figura 2.7 podemos ver el diagrama del data warehouse que involucra los datos de las fuentes 5 y 6 del área de Información Externa.

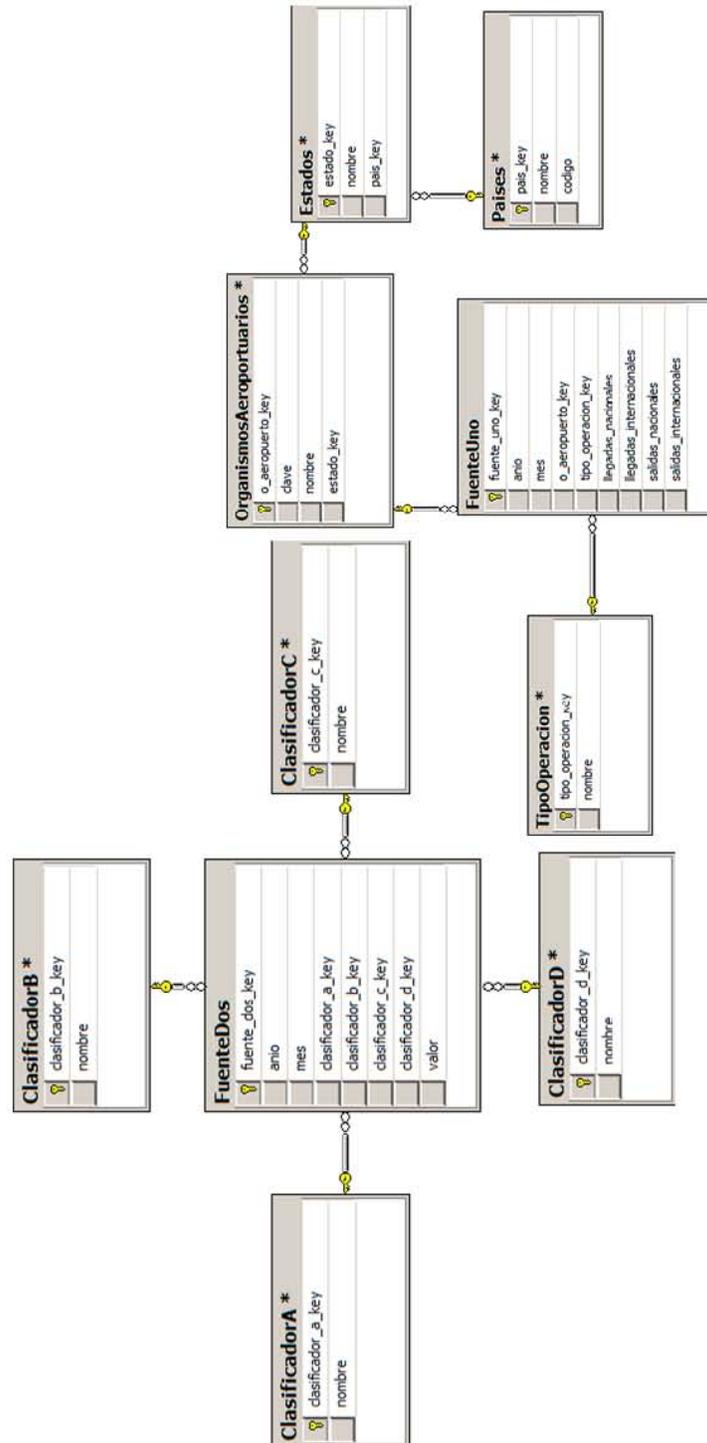


Figura 2.5: Diagrama del data warehouse, parte II

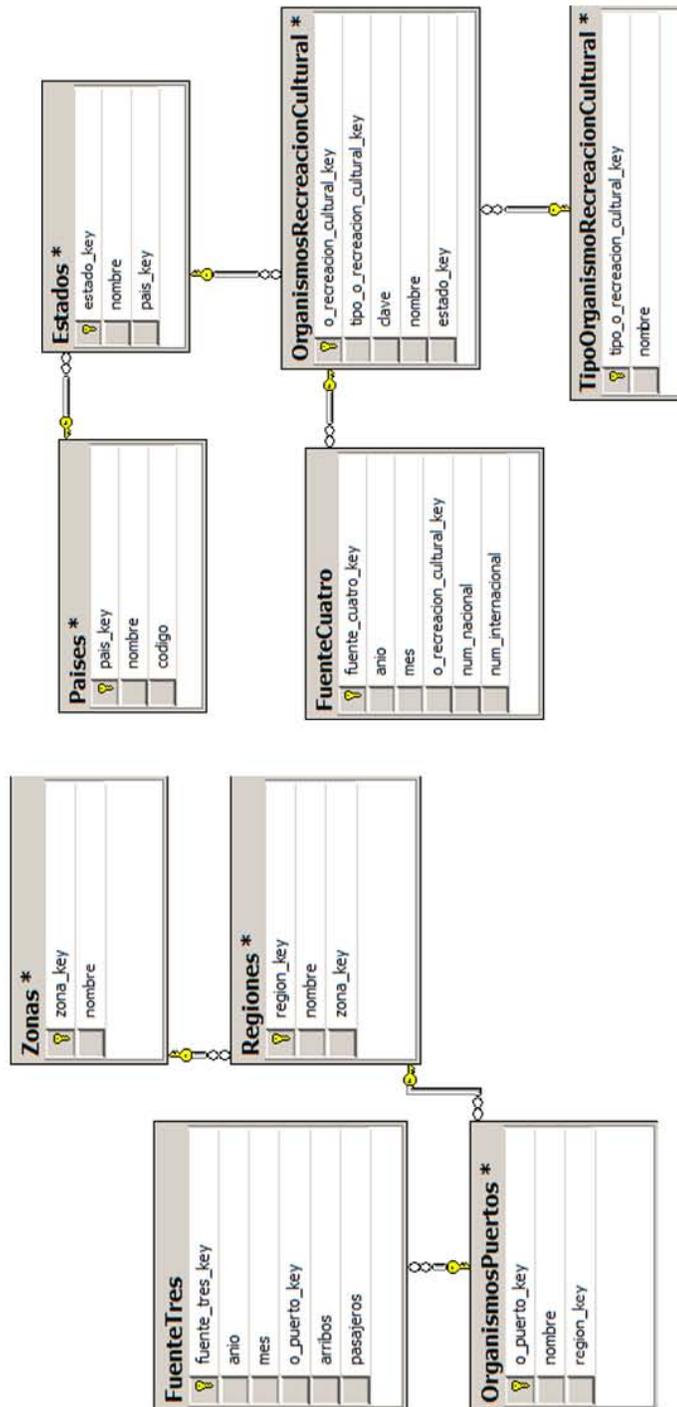


Figura 2.6: Diagrama del data warehouse, parte III

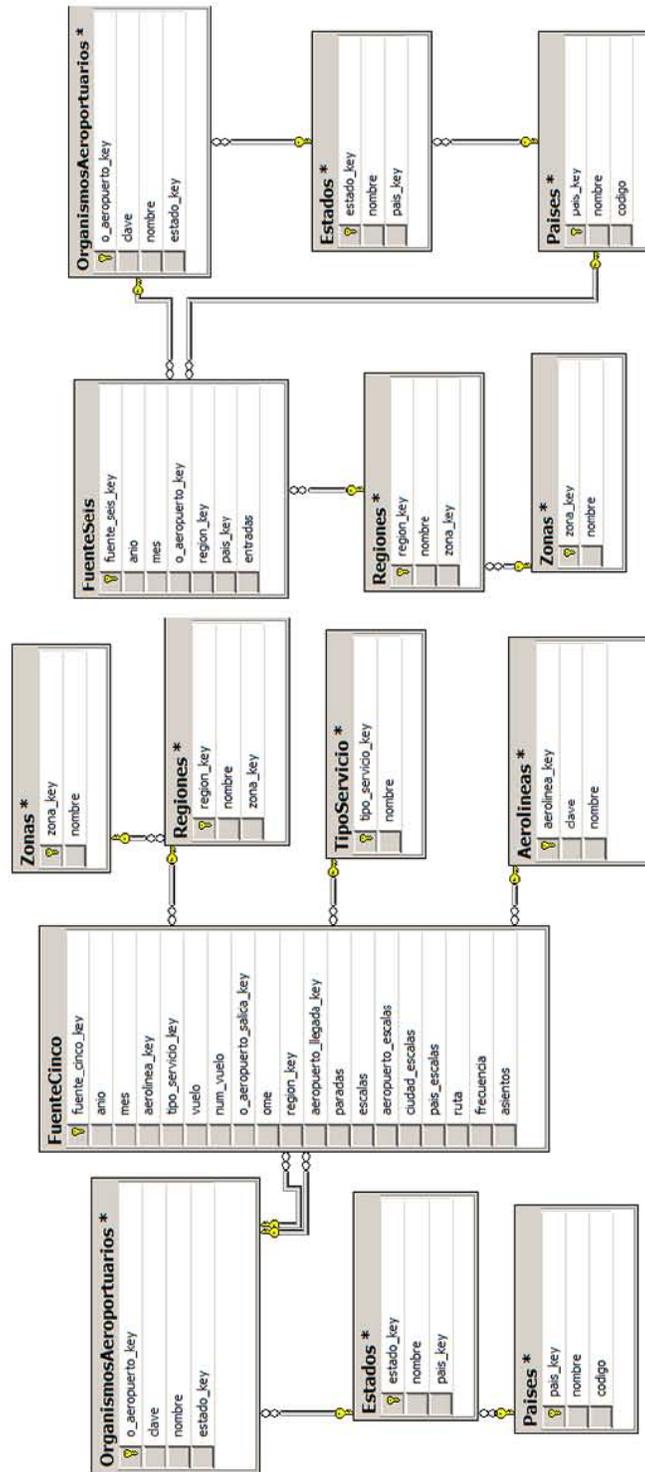


Figura 2.7: Diagrama del data warehouse, parte IV

2.3 Diseño de los ETL

En esta sección, se define qué es un ETL y, posteriormente, se presenta una descripción acerca de los seis tipos de fuentes a cargar en el sistema.

ETL⁷ es un proceso que permite mover datos desde múltiples fuentes, reformatear, limpiar, y cargar a los mismos en otra base de datos, data mart, o data warehouse. Estos procesos ETL tienen que traducir uno o varios sistemas operacionales normalizados e independientes a un único sistema, cuyos datos estén completamente integrados. El objetivo es preparar la información existente para su posterior análisis (ya sea por medio de OLAP o KDD⁸).

En el caso de este sistema, para desarrollar los procesos ETL había que fijarse en que los datos a cargar provenían de archivos de hoja de cálculo y de base de datos (.xls y .mdb, respectivamente), y debían guardarse en la base de datos operacional del sistema; posteriormente estos datos debían trasladarse al data warehouse.

Los archivos representaban datos que se recababan ya sea de manera mensual o semanal, por parte del cliente; sin embargo, el sistema debía poder hacer la carga de archivos cualquier día del año. Lo primero a realizar era cargar los datos a la base de datos y, ese mismo día por las noches, se debía reflejar esa carga en el data warehouse.

Mi aportación en el desarrollo de los procesos ETL fue el de diseñar e implementar un componente en el sistema, capaz de cargar los datos desde los archivos de hoja de cálculo en Microsoft Excel y de base de datos en Microsoft Access hacia la base de datos en SQL Server.

Teníamos de antemano, por parte de las reglas de negocio, que el formato de los archivos no había variado en los últimos diez años y que el mismo cliente veía muy poco probable que esto sucediera; es por eso que, se realizaron los procesos de

⁷ETL son las siglas en inglés de Extraer, Transformar y Cargar (Extract, Transform and Load)

⁸Descubrimiento de conocimiento en bases de datos (Knowledge Discovery in Databases), la cual se puede definir como: “Proceso no trivial de identificación de patrones valiosos, novedosos, potencialmente útiles y entendibles a partir de grandes volúmenes de datos”, U. Fayyad, et al. definición de KDD en el artículo KDD-96

*schema matching*⁹ y *schema mapping*¹⁰ de los datos fuente con respecto a la base de datos destino, de manera manual. Es decir, se identificaron los datos provenientes de los archivos que concordaban semánticamente con los datos de la base de datos y, posteriormente, se les aplicó una transformación para poderlos cargar al sistema.

A continuación se presentan los procesos de extracción, transformación y carga de datos, así como las consideraciones que se tomaron en cada uno de ellos.

Análisis

El tiempo de un proceso de ETL está marcado, entre otras cosas, por:

- El tamaño del archivo y la velocidad de subida en el sistema de origen (cliente).
- La velocidad de bajada y de procesamiento de datos en el sistema destino (servidor).

Se requirió que fuesen procesos que causaran el mínimo consumo de recursos tanto en el sistema de origen como en el de destino.

Extracción

Se contaba con grandes cantidades de archivos; el formato de cada uno de ellos había sido conservado durante varios años y, realmente, no se necesitaba de un nuevo diseño. Es por eso que se requirió hacer los procesos ETL a la medida, es decir, hacer que el sistema adoptase los formatos y dar por hecho que no cambiarían. Algunos archivos contienen un número indeterminado de filas, por lo que se optó por limitar la lectura de los datos a un número máximo de filas.

Transformación

Este proceso contempló el impacto de las reglas de negocio sobre los datos, así como una observación detallada sobre los mismos. Entre las observaciones encontradas, tenemos que:

⁹*Schema matching* se refiere al proceso de encontrar potenciales asociaciones entre elementos (por lo regular atributos o relaciones) de dos distintos esquemas de base de datos.

¹⁰*Schema mapping* es una especificación que describe cómo un conjunto de datos estructurados en un esquema de base de datos es transformado en otro conjunto de datos estructurados pero bajo un esquema diferente.

- Todos los archivos de las fuentes están expuestos a datos faltantes, en particular las fuentes 1 y 3.
- En general, el concepto de mes viene indicado en letra y en distintos formatos, cuándo en la base de datos debía guardarse con un número del 1 al 12.
- De igual manera, muchos de los organismos o referencias a localizaciones geográficas estaban marcadas con letra, cuando en la base de datos debía guardarse con número que indique su clave dentro del sistema.
- Todos los archivos tienen un formato ya definido por el cliente y se tienen datos ubicados en columnas fijas. Los únicos archivos con filas fijas son las fuentes 2 y 3.

Más adelante, en la sección de diseño del ETL, se presentarán en específico las transformaciones por cada dato a cargar dentro del archivo.

Carga

Por requerimiento del cliente, el proceso de carga debía seguir las siguientes reglas:

- Los datos ingresados que pertenezcan a un período donde ya se tienen registros, deberán reemplazar los existentes.
- Aquellos archivos que no cumplan con el formato definido para cada fuente serán descartados.
- Debido a que los datos contenidos en los archivos debían contener referencias a registros guardados en la base de datos, tenemos que, serán omitidos todos aquellos datos con referencias inexistentes en la base de datos.
- Se deberá guardar una bitácora con el registro de todos los archivos cargados, así como posibles errores, ligando al usuario responsable de cargar el archivo y fecha y hora en que se llevó a cabo.

Diseño

A continuación se muestra el caso de uso detallado para la carga de archivos en el sistema, por parte del área de Información Externa. Ver pantalla final en la Figura [A.1](#) del Apéndice [A](#).

Caso de uso: Carga de Información Externa.

Actor: Administrador

Descripción: Cargar en el sistema los archivos del área de Información Externa, involucrando procesos de ETL.

Pre-condición: El actor se ha autenticado.

Post-condición: El actor ha cargado su archivo al sistema

Tabla 2.2: Caso de uso Carga de Información Externa

Actor	Sistema	Excepciones
1. Entra a la aplicación	2. Muestra el menú, indicando los módulos permitidos al usuario	
3. Selecciona la opción <i>Carga de Información Externa</i>	4. Muestra una pantalla con seis pestañas, correspondientes a las seis distintas fuentes. Las etiquetas de las pestañas son: <i>Fuente 1</i> , <i>Fuente 2</i> , <i>Fuente 3</i> , <i>Fuente 4</i> , <i>Fuente 5</i> , <i>Fuente 6</i> . El contenido de las pestañas muestra: <ul style="list-style-type: none"> • un botón <i>Seleccionar archivo</i>. • un botón <i>Procesar</i>. • un botón <i>Cancelar</i>. La pestaña actual es la de <i>Fuente 1</i>	
5. Selecciona una pestaña, dependiendo la fuente que desee cargar	6. Muestra su contenido, descrito anteriormente	
7. Da click en el botón <i>Seleccionar archivo</i>	8. Muestra un pop-up donde pide al usuario el archivo a cargar, acompañado de los botones <i>Aceptar</i> y <i>Cancelar</i>	
9. Selecciona la ruta del archivo y da click en <i>Aceptar</i>	10. Oculta el pop-up e indica el nombre del archivo seleccionado	
11. Da click en <i>Procesar</i>	12. Obtiene de la máquina de origen el archivo y realiza un proceso de ETL, dependiendo el tipo de fuente seleccionada	E1

	13. Avisa al usuario el estatus de la operación procesada por el ETL y regresa al punto 4	E2
--	---	----

Tabla 2.3: Excepciones Caso de uso Carga de Información Externa

ID	Descripción
E1	El sistema indica que no se ha podido extraer el archivo. Se indican detalles acerca del error.
E2	El sistema indica que el archivo fue extraído, pero por problemas internos de procesamiento ha sido descartado. Se indican detalles acerca del error.

En cuanto al diseño de los algoritmos que se presentan a continuación, se tiene lo siguiente: los archivos debían procesarse en el sistema, guardarse en la base de datos y, posteriormente, una rutina por las noches debía actualizar los datos consolidados en el data warehouse. Esta rutina ya no hacía ningún tipo de limpieza, puesto que ya estaba hecha desde la base de datos; el data warehouse sólo actualiza sus datos a los que se tienen en la base de datos.

Se debía tener un control acerca de los archivos que entraban en la base de datos. La idea fue la siguiente: utilizar un lenguaje de programación que lleve el control del programa, reciba el archivo, lo guarde con la fecha y hora actual del sistema, extraiga los datos, identifique los datos requeridos para guardar, los valide, los transforme, y luego los inserte en la base de datos.

Para llevar a cabo la parte de extracción de los datos se debía recibir el archivo y acceder a los datos por medio de una conexión del lenguaje de programación hacia el archivo. Esta conexión debía permitir tanto consultas como modificaciones al archivo. De igual manera, en el proceso de carga, para guardar los datos en la base de datos había que utilizar una conexión a la base de datos.

Enseguida, se presentan los seis algoritmos utilizados en los procesos de ETL para las distintas fuentes.

Fuente 1**Input:** *archivo, ip_host, idUuario***Output:** *true, false***begin** *archivo.nombre* ← *archivo.nombre* + “-” + *getFechaHoraActual()* *rutaInicio* ← “home/archivosCargados/” *guardarArchivoEnRuta*(*archivo, rutaInicio*) *con1* ← *abrirConexionAExcel*(*archivo*) *con2* ← *abrirConexionBD*() *arregloDatos1* ← *cargarDatos*(*con1, “Pestana1”, “A6 : X256”*) **foreach** *fila* **in** *arregloDatos1* **do** *anio* ← *getColumna1*(*fila*) *nombreMes* ← *getColumna2*(*fila*), *mes* ← *pasarANumero*(*nombreMes*) *nombreAeropuerto* ← *getColumna4*(*fila*) *idA* ← *buscarIdEnBD*(*con2, “OrganismosAeroportuarios”, nombreAeropuerto*) *LNCR* ← *t*(*getColumna5*(*fila*)), *LICR* ← *t*(*getColumna6*(*fila*)) *sNCR* ← *t*(*getColumna7*(*fila*)), *sICR* ← *t*(*getColumna8*(*fila*)) *LNCH* ← *t*(*getColumna9*(*fila*)), *LICH* ← *t*(*getColumna10*(*fila*)) *sNCH* ← *t*(*getColumna11*(*fila*)), *sICH* ← *t*(*getColumna12*(*fila*)) *LNNR* ← *t*(*getColumna13*(*fila*)), *LINR* ← *t*(*getColumna14*(*fila*)) *sNNR* ← *t*(*getColumna15*(*fila*)), *sINR* ← *t*(*getColumna16*(*fila*)) *LNG* ← *t*(*getColumna17*(*fila*)), *LIG* ← *t*(*getColumna18*(*fila*)) *sNG* ← *t*(*getColumna19*(*fila*)), *sIG* ← *t*(*getColumna20*(*fila*)) **if** *formatoValidoF1*(*anio, mes, idA, LNCR, LICR, sNCR, sICR, LNCH, LICH,* *sNCH, sICH, LNNR, LINR, sNNR, sINR, LNG, LIG, sNG, sIG*) **then** *guardarEnTablaF1*(*con2, anio, mes, idA, “CR”, LNCR, LICR, sNCR, sICR*) *guardarEnTablaF1*(*con2, anio, mes, idA, “CH”, LNCH, LICH, sNCH, sICH*) *guardarEnTablaF1*(*con2, anio, mes, idA, “NR”, LNNR, LINR, sNNR, sINR*) *guardarEnTablaF1*(*con2, anio, mes, idA, “G”, LNG, LIG, sNG, sIG*) **else** *guardarEnBitacora*(*con2, “ETL F1”, archivo.nombre* + *getDescripcion*(*fila*), *getFechaHoraActual*(), “Error”, *ip_host, idUuario*) **end** **end** *guardarEnBitacora*(*con2, “ETL F1”, archivo.nombre,* *getFechaHoraActual*(), “Ok”, *ip_host, idUuario*) *cerrarConexionAExcel*(*con1*) *cerrarConexionBD*(*con2*) **return** *true***end****Algoritmo 1:** *etlFuente1*

Explicación del Algoritmo etlFuente1:

En el ETL de la Fuente 1, se recibe el archivo, se agrega la fecha y hora actual del sistema al nombre del archivo, y se guarda en la carpeta de archivos cargados. Posteriormente, se abren las conexiones a la base de datos y al archivo en Excel. Por medio de esta última, se cargan los datos del archivo de la Pestana1 del archivo desde la columna A a la X, y de la fila 6 a la 256. Esto es por que se tiene un número fijo de columnas, y por que los datos que se van a cargar comienzan desde la fila 6.

Además, ya que se tiene un número indeterminado de filas, se introduce como máximo 256 (ya que la regla de negocio nos dice que no hay más de 256 aeropuertos en México), tomando en cuenta que el número de filas es aproximadamente de 100 a 130 en los archivos.

Los datos se cargan en un arreglo bidimensional y se procede a recorrer el arreglo fila por fila. En cada iteración se obtiene la siguiente fila y se identifican los valores de las columnas por medio de las funciones *getColumnaK* y *t*. La función *getColumnaK* obtiene el valor de la columna *k*, para determinada fila. La función *t* toma el valor que se pasa por parámetro, y en caso de ser nulo o un símbolo “-”, regresa el valor 0 de manera predeterminada, en caso contrario, devuelve el mismo valor del parámetro.

En cuanto a los datos que se obtienen: el año proviene de la columna 1, el nombre del mes se obtiene de la columna 2, pero posteriormente se transforma a su número correspondiente del 1 al 12; la columna 3 se omite, el nombre del aeropuerto se obtiene de la columna 4, pero posteriormente se transforma a su clave equivalente en la base de datos por medio de la conexión y el nombre de la tabla “OrganismosAeroportuarios”; los datos correspondientes al tipo de operación “CR” se obtienen de las columnas 5, 6, 7 y 8; de manera análoga, los datos del tipo de operación “CH”, “NR”, y “G” se obtienen de las columnas (9,10,11,12), (13,14,15,16) y (17,18,19,20).

Si los datos obtenidos son válidos, es decir, el año es un número entero positivo de cuatro dígitos, el mes es un número entero con valor máximo 12 y mínimo 1, la clave del aeropuerto se encontró exitosamente en la base de datos, y los valores de los datos de “CR”, “CH”, “NR”, y “G” son números enteros positivos, entonces se procede a guardar la fila en 4 registros para la tabla “FuenteUno” en la base de datos; son cuatro ya que es un registro por cada tipo de operación. En caso que la

fila contenga algún valor no válido, entonces se descarta y se guarda un registro en la bitácora, indicando que se trataba de la carga de un archivo para la Fuente 1, el nombre del archivo, las columnas de la fila, la fecha y hora del sistema, el motivo del descarte, la ip y clave del usuario.

Al final, de igual manera, se guarda en el registro de la bitácora que se ha cargado el archivo al sistema.

Fuente 2

```

Input: archivo, ip_host, idUuario
Output: true, false
begin
  archivo.nombre ← archivo.nombre + "_" + getFechaHoraActual()
  rutaInicio ← "home/archivosCargados/"
  guardarArchivoEnRuta(archivo, rutaInicio)
  con1 ← abrirConexionAExcel(archivo)
  con2 ← abrirConexionBD()
  arregloDatos1 ← cargarDatos(con1, "Pestana1", "B13, B14, B16, B17, B21, B22, B29,
  B30, B32, B33, B37, B38, E13, E14, E16, E17, E21, E22, E29, E30, E32, E33, E37, E38")
  celdaAnioMes ← cargarDatos(con1, "Pestana1", "B3")
  anio ← getAnio(celdaAnioMes)
  mes ← getMes(celdaAnioMes)
  if ¬ formatoAnioMesValido(anio, mes) then
    guardarEnBitacora(con2, "ETL F2", archivo.nombre +
    getDescripcion(anio, mes), getFechaHoraActual(), "Error", ip_host, idUuario)
    cerrarConexionAExcel(con1)
    cerrarConexionBD(con2)
    return false
  end
  guardarEnTablaF2(con2, anio, mes, "IN", "TI", "TDI", "VA", t(getCeldaB13(arregloDatos1)))
  guardarEnTablaF2(con2, anio, mes, "IN", "TI", "TDI", "VT", t(getCeldaB14(arregloDatos1)))
  guardarEnTablaF2(con2, anio, mes, "IN", "TI", "TF", "P", t(getCeldaB16(arregloDatos1)))
  guardarEnTablaF2(con2, anio, mes, "IN", "TI", "TF", "E", t(getCeldaB17(arregloDatos1)))
  guardarEnTablaF2(con2, anio, mes, "IN", "EI", "EF", "P", t(getCeldaB21(arregloDatos1)))
  guardarEnTablaF2(con2, anio, mes, "IN", "EI", "EF", "E", t(getCeldaB22(arregloDatos1)))
  guardarEnTablaF2(con2, anio, mes, "EG", "TI", "TDI", "VA", t(getCeldaB29(arregloDatos1)))
  guardarEnTablaF2(con2, anio, mes, "EG", "TI", "TDI", "VT", t(getCeldaB30(arregloDatos1)))
  guardarEnTablaF2(con2, anio, mes, "EG", "TI", "TF", "P", t(getCeldaB32(arregloDatos1)))
  guardarEnTablaF2(con2, anio, mes, "EG", "TI", "TF", "E", t(getCeldaB33(arregloDatos1)))
  guardarEnTablaF2(con2, anio, mes, "EG", "EI", "EF", "P", t(getCeldaB37(arregloDatos1)))
  guardarEnTablaF2(con2, anio, mes, "EG", "EI", "EF", "E", t(getCeldaB38(arregloDatos1)))
  guardarEnTablaF2(con2, anio, mes, "GMIN", "TI", "TDI", "VA", t(getCeldaE13(arregloDatos1)))
  guardarEnTablaF2(con2, anio, mes, "GMIN", "TI", "TDI", "VT", t(getCeldaE14(arregloDatos1)))
  guardarEnTablaF2(con2, anio, mes, "GMIN", "TI", "TF", "P", t(getCeldaE16(arregloDatos1)))
  guardarEnTablaF2(con2, anio, mes, "GMIN", "TI", "TF", "E", t(getCeldaE17(arregloDatos1)))
  guardarEnTablaF2(con2, anio, mes, "GMIN", "EI", "EF", "P", t(getCeldaE21(arregloDatos1)))
  guardarEnTablaF2(con2, anio, mes, "GMIN", "EI", "EF", "E", t(getCeldaE22(arregloDatos1)))
  guardarEnTablaF2(con2, anio, mes, "GEEG", "TI", "TDI", "VA", t(getCeldaE29(arregloDatos1)))
  guardarEnTablaF2(con2, anio, mes, "GEEG", "TI", "TDI", "VT", t(getCeldaE30(arregloDatos1)))
  guardarEnTablaF2(con2, anio, mes, "GEEG", "TI", "TF", "P", t(getCeldaE32(arregloDatos1)))
  guardarEnTablaF2(con2, anio, mes, "GEEG", "TI", "TF", "E", t(getCeldaE33(arregloDatos1)))
  guardarEnTablaF2(con2, anio, mes, "GEEG", "EI", "EF", "P", t(getCeldaE37(arregloDatos1)))
  guardarEnTablaF2(con2, anio, mes, "GEEG", "EI", "EF", "E", t(getCeldaE38(arregloDatos1)))
  guardarEnBitacora(con2, "ETL F2", archivo.nombre,
  getFechaHoraActual(), "Ok", ip_host, idUuario)
  cerrarConexionAExcel(con1)
  cerrarConexionBD(con2)
  return true
end

```

Algoritmo 2: etlFuente2

Explicación del Algoritmo etlFuente2:

En el ETL de la Fuente 2, se recibe el archivo, se agrega la fecha y hora actual del

sistema al nombre del archivo, y se guarda en la carpeta de archivos cargados. Posteriormente, se abren las conexiones a la base de datos y al archivo en Excel. Por medio de esta última, se cargan los datos de la Pestana1 del archivo, desde 24 celdas en específico: “B13, B14, B16, B17, B21, B22, B29, B30, B32, B33, B37, B38, E13, E14, E16, E17, E21, E22, E29, E30, E32, E33, E37, E38”.

Las celdas seleccionadas hacen referencia a los datos primarios de los 4 bloques que hay en la parte de arriba del archivo. Esto es porque todos los demás datos que hay se derivan de éstos, y ya que se trata de guardarlos en la base de datos, se busca guardar lo mínimo pero lo suficiente (la rutina por las noches que carga los datos en el data warehouse hará lo necesario para recuperar los datos relevantes para el cliente, a partir de éstos).

Los datos se cargan en un arreglo de tamaño 24. Además, se puede obtener directamente el año y el mes (número) desde la celda B3. Si el año es un número entero positivo de 4 dígitos y el mes es un número entero con valor máximo 12 y mínimo 1, entonces se procede a obtener los datos del arreglo. Por cada dato se hace una inserción para la tabla “FuenteDos” en la base de datos. Se trata de inserciones identificadas manualmente, es decir, a partir de los encabezados en el archivo de Excel se le pasan como parámetros el año, el mes, 4 clasificadores y el valor de la celda.

El valor de la celda se extrae por medio de las funciones *getCeldaJ* y *t2*. La función *getCeldaJ* devuelve el valor de la celda *J* para determinado arreglo. La función *t* es la misma que la descrita en la Fuente 1. Todos los datos del arreglo deben guardarse.

Fuente 3**Input:** *archivo, ip_host, idUsuario***Output:** *true, false***begin** *archivo.nombre* ← *archivo.nombre* + “-” + *getFechaHoraActual()* *rutaInicio* ← “home/archivosCargados/” *guardarArchivoEnRuta*(*archivo, rutaInicio*) *con1* ← *abrirConexionAExcel*(*archivo*) *con2* ← *abrirConexionBD*() *arregloDatos1* ← *cargarDatos*(*con1, “Pestana1”, “A10 : C27, A31 : C41”*) *anio* ← *cargarDatos*(*con1, “Pestana1”, “A2”*) *nombreMes* ← *cargarDatos*(*con1, “Pestana1”, “B2”*) *mes* ← *pasarANumero*(*nombreMes*) **if** ¬ *formatoAnioMesValido*(*anio, mes*) **then** *guardarEnBitacora*(*con2, “ETL F3”, archivo.nombre* + *getDescripcion*(*anio, mes*), *getFechaHoraActual()*, “Error”, *ip_host, idUsuario*) *cerrarConexionAExcel*(*con1*) *cerrarConexionBD*(*con2*) **return** *false* **end** **foreach** *fila* **in** *arregloDatos1* **do** *nombrePuerto* ← *t*(*getColumna1*(*fila*)) *idP* ← *buscarIdEnBD*(*con2, “OrganismosPuertos”, nombrePuerto*) *numArribos* ← *t*(*getColumna2*(*fila*)) *numPasajeros* ← *t*(*getColumna3*(*fila*)) **if** *formatoValidoF3*(*idP, numArribos, numPasajeros*) **then** *guardarEnTablaF3*(*con2, anio, mes, idP, numArribos, numPasajeros*) **else** *guardarEnBitacora*(*con2, “ETL F3”, archivo.nombre* + *getDescripcion*(*fila*), *getFechaHoraActual()*, “Error”, *ip_host, idUsuario*) **end** **end** *guardarEnBitacora*(*con2, “ETL F3”, archivo.nombre,* *getFechaHoraActual()*, “Ok”, *ip_host, idUsuario*) *cerrarConexionAExcel*(*con1*) *cerrarConexionBD*(*con2*) **return** *true***end****Algoritmo 3:** *etlFuente3*

Explicación del Algoritmo etlFuente3:

En el ETL de la Fuente 3, se recibe el archivo, se agrega la fecha y hora actual del sistema al nombre del archivo, y se guarda en la carpeta de archivos cargados. Posteriormente, se abren las conexiones a la base de datos y al archivo en Excel. Por medio de esta última, se cargan los datos de la Pestaña1 del archivo, desde dos grupos de de celdas. La primera, compuesta desde la celda superior izquierda A10 a la celda inferior derecha C27. La segunda, desde la celda superior izquierda A31 a la celda inferior derecha C41. Esto es por que se tiene un número fijo de columnas (3), y por que los datos que se van a cargar comienzan desde la fila 10 hasta la 41, omitiendo las filas 28,29,30.

Los datos se cargan en un arreglo bidimensional. Además, se puede obtener directamente el año y el mes (número) desde las celdas A2 y B2. Si el año es un número entero positivo de 4 dígitos y el mes es un número entero con valor máximo 12 y mínimo 1, entonces se procede a recorrer el arreglo fila por fila. En cada iteración se obtiene la siguiente fila y se identifican los valores de las columnas por medio de las funciones *getColumnaK* y *t*, descritas en la explicación del ETL para la Fuente 1.

En cuanto a los datos que se obtienen: el nombre del puerto se obtiene de la columna 1, pero posteriormente se transforma a su clave equivalente en la base de datos por medio de la conexión y el nombre de la tabla “OrganismosPuertos”; también se obtienen los números de arribos y de pasajeros para ese puerto en las columnas 2 y 3, respectivamente.

Si los datos obtenidos son válidos, es decir, la clave del puerto se encontró exitosamente en la base de datos, y los valores de los arribos y pasajeros son números enteros positivos, entonces se procede a guardar la fila en un registro para la tabla “FuenteTres” en la base de datos. En caso que la fila contenga algún valor no válido, entonces se descarta y se guarda un registro en la bitácora, indicando que se trataba de la carga de un archivo para la Fuente 3, el nombre del archivo, las columnas de la fila, la fecha y hora del sistema, el motivo del descarte, la ip y clave del usuario. Al final, de igual manera, se guarda en el registro de la bitácora que se ha cargado el archivo al sistema.

Fuente 4**Input:** *archivo, ip_host, idUuario***Output:** *true, false***begin** *archivo.nombre* ← *archivo.nombre* + “-” + *getFechaHoraActual()* *rutaInicio* ← “home/archivosCargados/” *guardarArchivoEnRuta*(*archivo, rutaInicio*) *con1* ← *abrirConexionAExcel*(*archivo*) *con2* ← *abrirConexionBD*() *arregloDatos1* ← *cargarDatos*(*con1, “Pestana1”, “A3 : F256”*) *celdaAnioMes* ← *cargarDatos*(*con1, “Pestana1”, “A2”*) *anio* ← *getAnio*(*celdaAnioMes*) *mes* ← *getMes*(*celdaAnioMes*) **if** ¬ *formatoAnioMesValido*(*anio, mes*) **then** *guardarEnBitacora*(*con2, “ETL F4”, archivo.nombre* + *getDescripcion*(*anio, mes*), *getFechaHoraActual()*, “Error”, *ip_host, idUuario*) *cerrarConexionAExcel*(*con1*) *cerrarConexionBD*(*con2*) **return** *false* **end** *idPais* ← *buscarIdEnBD*(*con2, “Paises”, “Mexico”*) **foreach** *fila* in *arregloDatos1* **do** *nombreEstado* ← *t*(*getColumna1*(*fila*)) *idE* ← *buscarIdEnBD*(*con2, “Estados”, nombreEstado, idPais*) *tipoOrganismo* ← *t*(*getColumna2*(*fila*)) *idTO* ← *buscarIdEnBD*(*con2, “TipoOrganismoRecreacionCultural”, tipoOrganismo*) *nombreOrganismo* ← *t*(*getColumna3*(*fila*)) *idO* ← *buscarIdEnBD*(*con2, “OrganismosRecreacionCultural”, nombreOrganismo, idE, idTO*) *numNacionales* ← *t*(*getColumna4*(*fila*)) *numExtranjeros* ← *t*(*getColumna5*(*fila*)) **if** *formatoValidoF4*(*idO, numNacionales, numExtranjeros*) **then** *guardarEnTablaF4*(*con2, anio, mes, idO, numNacionales, numExtranjeros*) **else** *guardarEnBitacora*(*con2, “ETL F4”, archivo.nombre* + *getDescripcion*(*fila*), *getFechaHoraActual()*, “Error”, *ip_host, idUuario*) **end** **end** *guardarEnBitacora*(*con2, “ETL F4”, archivo.nombre,* *getFechaHoraActual()*, “Ok”, *ip_host, idUuario*) *cerrarConexionAExcel*(*con1*) *cerrarConexionBD*(*con2*) **return** *true***end****Algoritmo 4:** *etlFuente4*

Explicación del Algoritmo etlFuente4:

En el ETL de la Fuente 4, se recibe el archivo, se agrega la fecha y hora actual del sistema al nombre del archivo, y se guarda en la carpeta de archivos cargados. Posteriormente, se abren las conexiones a la base de datos y al archivo en Excel. Por medio de esta última, se cargan los datos del archivo de la Pestana1 del archivo desde la columna A a la F, y de la fila 3 a la 256. Esto es por que se tiene un número fijo de columnas, y por que los datos que se van a cargar comienzan desde la fila 6. Además, ya que se tiene un número indeterminado de filas, se introduce como máximo 256, tomando en cuenta que el número de filas es aproximadamente de 100 a 130 en los archivos.

Los datos se cargan en un arreglo bidimensional. Además, se puede obtener directamente el año y el mes (número) desde la celda A2. Si el año es un número entero positivo de 4 dígitos y el mes es un número entero con valor máximo 12 y mínimo 1, entonces continua. Se obtiene el valor de la clave del país México en la base de datos y luego se procede a recorrer el arreglo fila por fila. En cada iteración se obtiene la siguiente fila y se identifican los valores de las columnas por medio de las funciones *getColumnaK* y *t*, descritas en la explicación del ETL para la Fuente 1.

En cuanto a los datos que se obtienen: el nombre de la Entidad Federativa, el tipo de Organismo y el mismo Organismo se obtienen de las columnas 1,2 y 3, pero posteriormente se transforman a sus claves equivalentes en la base de datos por medio de la conexión y el nombre de las tablas “Estados”, “TipoOrganismoRecreacionCultural” y “OrganismosRecreacionCultural”; también se obtienen números acerca visitas nacionales y extranjeras para ese organismos en las columnas 4 y 5, respectivamente.

Si los datos obtenidos son válidos, es decir, la clave del organismo se encontró exitosamente en la base de datos, y los valores de visitas nacionales y extranjeras son números enteros positivos, entonces se procede a guardar la fila en un registro para la tabla “FuenteCuatro” en la base de datos. En caso que la fila contenga algún valor no válido, entonces se descarta y se guarda un registro en la bitácora, indicando que se trataba de la carga de un archivo para la Fuente 4, el nombre del archivo, las columnas de la fila, la fecha y hora del sistema, el motivo del descarte, la ip y clave del usuario. Al final, de igual manera, se guarda en el registro de la

bitácora que se ha cargado el archivo al sistema.

Fuente 5

```

Input: archivo, ip_host, idUuario
Output: true, false
begin
  archivo.nombre ← archivo.nombre + "_" + getFechaHoraActual()
  rutaInicio ← "home/archivosCargados/"
  guardarArchivoEnRuta(archivo, rutaInicio)
  con1 ← abrirConexionAccess(archivo)
  con2 ← abrirConexionBD()
  index ← 0, inc ← 500
  numRegistros ← cargarDatos(con1, "SELECT COUNT(*) FROM TABLA_FUENTE_5")
  while index ≤ numRegistros do
    arregloDatos1 ← cargarDatos(con1, "SELECT * FROM TABLA_FUENTE_5 WHERE id >= "
    + index + " AND id <=" + (index+inc)
    foreach fila in arregloDatos1 do
      anio ← t(getColumna1(fila))
      nombreMes ← t(getColumna2(fila)), mes ← pasarANumero(nombreMes)
      tipoServicio ← t(getColumna3(fila))
      idTipoServicio ← buscarIdEnBD(con2, "TipoServicio", tipoServicio)
      vuelo ← t(getColumna5(fila))
      claveAerolinea ← t(getColumna6(fila))
      idAerolinea ← buscarIdEnBD(con2, "Aerolineas", claveAerolinea)
      numVuelo ← t(getColumna8(fila))
      claveAeropuertoSalida ← t(getColumna9(fila))
      idAeropuertoSalida ←
      buscarIdEnBD(con2, "OrganismosAeroportuarios", claveAeropuertoSalida)
      ome ← t(getColumna13(fila))
      nombreRegion ← t(getColumna14(fila))
      idRegion ← buscarIdEnBD(con2, "Regiones", nombreRegion)
      claveAeropuertoLlegada ← t(getColumna16(fila))
      idAeropuertoLlegada ← buscarIdEnBD(con2, "Aeropuertos", claveAeropuertoLlegada)
      paradas ← t(getColumna20(fila))
      escalas ← t(getColumna21(fila))
      aeropuertoEscalas ← t(getColumna22(fila))
      ciudadEscalas ← t(getColumna23(fila))
      paisEscalas ← t(getColumna24(fila))
      ruta ← t(getColumna25(fila))
      frecuencia ← t(getColumna26(fila))
      asientos ← t(getColumna27(fila))
      if formatoValidoF5(anio, mes, idTipoServicio, vuelo, idAerolinea, numVuelo,
      idAeropuertoSalida, ome, idRegion, idAeropuertoLlegada, paradas, escalas,
      aeropuertoEscalas, ciudadEscalas, paisEscalas, ruta, frecuencia, asientos) then
        guardarEnTablaF5(con2, anio, mes, idTipoServicio, vuelo, idAerolinea, numVuelo,
        idAeropuertoSalida, ome, idRegion, idAeropuertoLlegada, paradas, escalas,
        aeropuertoEscalas, ciudadEscalas, paisEscalas, ruta, frecuencia, asientos)
      else
        guardarEnBitacora(con2, "ETL F5", archivo.nombre +
        getDescripcion(fila), getFechaHoraActual(), "Error", ip_host, idUuario)
      end
    end
    index ← index + inc
  end
  guardarEnBitacora(con2, "ETL F5", archivo.nombre,
  getFechaHoraActual(), "Ok", ip_host, idUuario)
  cerrarConexionAccess(con1)
  cerrarConexionBD(con2)
  return true
end

```

Algoritmo 5: etlFuente5

Explicación del Algoritmo etlFuente5:

En el ETL de la Fuente 5, se recibe el archivo, se agrega la fecha y hora actual del sistema al nombre del archivo, y se guarda en la carpeta de archivos cargados. Posteriormente, se abren las conexiones a la base de datos y al archivo en Access. Por medio de esta última, con ayuda de dos *scripts* en SQL, un ciclo y dos variables más, se cargan los datos del archivo en grupos de 500 registros, para poder procesarlos sin saturar la memoria.

Los datos se cargan en un arreglo bidimensional y se recorre fila por fila. En cada iteración se obtiene la siguiente fila y se identifican los valores de las columnas por medio de las funciones *getColumnaK* y *t*, descritas en la explicación del algoritmo para la Fuente 1.

En cuanto a los datos que se obtienen: el año proviene de la columna 1, el nombre del mes se obtiene de la columna 2, pero posteriormente se transforma a su número correspondiente del 1 al 12; el nombre del tipo de servicio, aerolínea, aeropuerto de salida, región y aeropuerto de llegada se obtienen de las columnas 3,6,9,14 y 16, respectivamente, pero luego se transforman a sus claves equivalentes en la base de datos por medio de la conexión y el nombre de las tablas “TipoServicio”, “Aerolíneas”, “Regiones” y “OrganismosAeroportuarios”; también se obtienen datos acerca del vuelo, número de vuelo, ome, número de paradas, escalas, claves de los aeropuertos, ciudades y países de las escalas, además de la ruta, el número de frecuencia y el de asientos de las columnas 5,8,13,20,21,22,23,24,25,26 y 27, respectivamente. Las columnas 4,7,10,11,12,15,17,18,19 se omiten.

Si los datos obtenidos son válidos, es decir, las claves del tipo de servicio, aerolínea, aeropuerto de salida, región y aeropuerto de llegada se encontraron exitosamente en la base de datos; los valores del vuelo, número de vuelo, paradas, frecuencia y asientos son números enteros positivos; y los valores de ome, escalas, ruta, claves de los aeropuertos, ciudades y países de las escalas son cadenas de caracteres; entonces se procede a guardar la fila en un registro para la tabla “FuenteCinco” en la base de datos. En caso que la fila contenga algún valor no válido, entonces se descarta y se guarda un registro en la bitácora, indicando que se trataba de la carga de un archivo para la Fuente 5, el nombre del archivo, las columnas de la fila, la fecha y hora del sistema, el motivo del descarte, la ip y clave del usuario. Al final, de igual manera, se guarda en el registro de la bitácora que se ha cargado el archivo al

sistema.

Fuente 6**Input:** *archivo, ip_host, idUsuario***Output:** *true, false***begin** *archivo.nombre* ← *archivo.nombre* + “-” + *getFechaHoraActual()* *rutaInicio* ← “home/archivosCargados/” *guardarArchivoEnRuta*(*archivo, rutaInicio*) *con1* ← *abrirConexionAccess*(*archivo*) *con2* ← *abrirConexionBD*() *index* ← 0, *inc* ← 500 *numRegistros* ← *cargarDatos*(*con1*, “SELECT COUNT(*) FROM TABLA_FUENTE_6”) **while** *index* ≤ *numRegistros* **do** *arregloDatos1* ← *cargarDatos*(*con1*, “SELECT * FROM TABLA_FUENTE_6 WHERE id ≥ ” + *index* + “ AND id ≤ ” + (*index*+*inc*)) **foreach** *fila* **in** *arregloDatos1* **do** *anio* ← *t*(*getColumna1*(*fila*)) *mes* ← *t*(*getColumna2*(*fila*)) *hub* ← *t*(*getColumna3*(*fila*)) *idAeropuerto* ← *buscarIdEnBD*(*con2*, “OrganismosAeroportuarios”, *hub*) *nombreRegion* ← *t*(*getColumna4*(*fila*)) *idRegion* ← *buscarIdEnBD*(*con2*, “Regiones”, *nombreRegion*) *nombrePais* ← *t*(*getColumna5*(*fila*)) *idPais* ← *buscarIdEnBD*(*con2*, “Paises”, *nombrePais*) *entradas* ← *t*(*getColumna6*(*fila*)) **if** *formatoValidoF6*(*anio, mes, idAeropuerto, idRegion, idPais, entradas*) **then** | *guardarEnTablaF6*(*anio, mes, idAeropuerto, idRegion, idPais, entradas*) **else** | *guardarEnBitacora*(*con2*, “ETL F6”, *archivo.nombre* + | *getDescripcion*(*fila*), *getFechaHoraActual*(), “Error”, *ip_host, idUsuario*) **end** **end** *index* ← *index* + *inc* **end** *guardarEnBitacora*(*con2*, “ETL F6”, *archivo.nombre*, *getFechaHoraActual*(), “Ok”, *ip_host, idUsuario*) *cerrarConexionAccess*(*con1*) *cerrarConexionBD*(*con2*) **return** *true***end****Algoritmo 6:** *etlFuente6*

Explicación del Algoritmo etlFuente6:

En el ETL de la Fuente 6, se recibe el archivo, se agrega la fecha y hora actual del sistema al nombre del archivo, y se guarda en la carpeta de archivos cargados. Posteriormente, se abren las conexiones a la base de datos y al archivo en Access. Por medio de esta última, con ayuda de dos *scripts* en SQL, un ciclo y dos variables más, se cargan los datos del archivo en grupos de 500 registros, para poder procesarlos sin saturar la memoria.

Los datos se cargan en un arreglo bidimensional y se recorre fila por fila. En cada iteración se obtiene la siguiente fila y se identifican los valores de las columnas por medio de las funciones *getColumnaK* y *t*, descritas en la explicación del algoritmo para la Fuente 1.

En cuanto a los datos que se obtienen: el año proviene de la columna 1, el nombre del mes se obtiene de la columna 2, pero posteriormente se transforma a su número correspondiente del 1 al 12; el nombre del aeropuerto, la región y el país se obtienen de las columnas 1,2 y 3, respectivamente, pero luego se transforman a sus claves equivalentes en la base de datos por medio de la conexión y el nombre de las tablas “OrganismosAeroportuarios”, “Regiones” y “Paises”; también se obtienen datos acerca del número de entradas de las columna 6.

Si los datos obtenidos son válidos, es decir, las claves aeropuerto, región y país se encontraron exitosamente en la base de datos, y el valor de las entradas es un número entero positivo, entonces se procede a guardar la fila en un registro para la tabla “FuenteSeis” en la base de datos. En caso que la fila contenga algún valor no válido, entonces se descarta y se guarda un registro en la bitácora, indicando que se trataba de la carga de un archivo para la Fuente 6, el nombre del archivo, las columnas de la fila, la fecha y hora del sistema, el motivo del descarte, la ip y clave del usuario. Al final, de igual manera, se guarda en el registro de la bitácora que se ha cargado el archivo al sistema.

2.4 Diseño del formulario con validación al vuelo

Análisis

A continuación se describe la estrategia a seguir para la implementación de validaciones *al vuelo* del formulario vía página web.

Lo primero que se hizo fue transcribir las reglas de negocio a su forma algebraica. Recordando las reglas descritas en el capítulo de Requerimientos del sistema (y renombrando las variables Dato1 como los cuartos registrados, así como siete columnas que representan los cuartos disponibles, cuartos ocupados por residentes, cuartos ocupados por no residentes, llegada de turistas residentes, llegada de turistas no residentes, turistas noche de residentes y turistas noche de no residentes, en ese orden), tenemos la siguiente tabla:

Tabla 2.4: Reglas de validación en la captura de Información Básica

Regla	Descripción
Regla 0	Los valores de las celdas de las siete columnas deben ser números enteros positivos
Regla 1	El valor de la celda de la columna 1 debe ser menor o igual al valor del Dato1. Este Dato1 se refiere a un atributo que tiene el organismo y que es mayor a cero.
Regla 2	El valor de la celda de la columna 2 debe ser menor o igual al valor de la celda de la columna 1
Regla 3	El valor de la celda de la columna 3 debe ser menor o igual al valor de la celda de la columna 1
Regla 4	La suma de los valores de las celdas de las columnas 2 y 3 debe ser menor o igual al valor de la celda de la columna 1
Regla 5	El valor de la celda de la columna 6 debe ser menor o igual a la multiplicación del valor de la celda de la columna 2 por una constante igual a 8
Regla 6	El valor de la celda de la columna 4 debe ser menor o igual al valor de la celda de la columna 6
Regla 7	El valor de la celda de la columna 7 debe ser menor o igual a la multiplicación del valor de la celda de la columna 3 por una constante igual a 8

Regla 8	El valor de la celda de la columna 5 debe ser menor o igual al valor de la celda de la columna 7
Regla 9	El valor de la celda de la columna 6 debe ser menor o igual que la suma del valor de la celda de la columna 4 más el valor del día anterior en la celda de la columna 6. En el caso del primer día, se debe tomar como valor del día anterior por defecto el número 0
Regla 10	El valor de la celda de la columna 7 debe ser menor o igual que la suma del valor de la celda de la columna 5 más el valor del día anterior en la celda de la columna 7. En el caso del primer día, se debe tomar como valor del día anterior por defecto el número 0

En la Figura 2.8 podemos ver el prototipo del cuestionario y los campos relacionados con la validación de las diez reglas (en este caso para el día dos).

El prototipo muestra una interfaz de usuario dividida en dos secciones principales: 'Datos organismo' y 'Cuestionario'.

En la sección 'Datos organismo', hay un campo 'Nombre organismo' con el valor 'Organismo1' y un campo 'Dato1' con el valor 'Valor1'. Una flecha roja etiquetada como 'Regla 1' apunta desde 'Dato1' hacia el cuestionario.

La sección 'Cuestionario' es una tabla con 7 columnas (Columna1 a Columna7) y 3 filas (Dia 1, Dia 2, Dia 3). Cada celda contiene un campo de entrada de texto.

Las reglas de validación se representan por flechas de colores que conectan reglas con campos específicos:

- Regla 1 (rojo):** apunta a la celda (Dia 2, Columna1).
- Regla 2 (azul):** apunta a la celda (Dia 2, Columna2).
- Regla 3 (naranja):** apunta a la celda (Dia 2, Columna3).
- Regla 4 (magenta):** apunta a la celda (Dia 2, Columna4).
- Regla 5 (verde):** apunta a la celda (Dia 2, Columna5).
- Regla 6 (azul claro):** apunta a la celda (Dia 2, Columna6).
- Regla 7 (naranja claro):** apunta a la celda (Dia 2, Columna7).
- Regla 8 (naranja):** apunta a la celda (Dia 2, Columna5).
- Regla 9 (verde claro):** apunta a la celda (Dia 2, Columna6).
- Regla 10 (púrpura):** apunta a la celda (Dia 2, Columna7).

Además, hay flechas verdes que indican dependencias entre días: Regla 5 apunta de Dia 1 a Dia 2 en Columna5; Regla 9 apunta de Dia 1 a Dia 2 en Columna6; Regla 10 apunta de Dia 1 a Dia 2 en Columna7.

En la parte inferior del cuestionario, hay tres botones: 'Guardar', 'Limpiar' y 'Cancelar'.

Figura 2.8: Prototipo captura de Información Básica

Teniendo en cuenta lo anterior, sea:

- σ el valor del Dato1, atributo del organismo.
- c_i el valor de la celda de la i -ésima columna del día a validar.

- C_i^* el valor de la celda de la i -ésima columna del día anterior.
- $\kappa = 8$

Las reglas quedan:

- $R_0 : \forall c_i, c_i \in \mathbb{Z}^+$
- $R_1 : c_1 \leq \sigma$
- $R_2 : c_2 \leq c_1$
- $R_3 : c_3 \leq c_1$
- $R_4 : c_2 + c_3 \leq c_1$
- $R_5 : c_6 \leq c_2 * \kappa$
- $R_6 : c_4 \leq c_6$
- $R_7 : c_7 \leq c_3 * \kappa$
- $R_8 : c_5 \leq c_7$
- $R_9 : c_6 \leq c_4 + C_6^*$
- $R_{10} : c_7 \leq c_5 + C_7^*$

Para hacer las validaciones de este cuestionario, se tomó en cuenta la forma natural en que el usuario completa los datos, y esto es de izquierda a derecha. El usuario tratará de capturar el cuestionario en el siguiente orden: columna 1, columna 2, columna 3, columna 4, columna 5, columna 6 y columna 7.

Por otra parte, analizando las dependencias entre reglas, podemos ver que:

- Para validar la regla 2 y la regla 3 antes falta validar la regla 1,

pues en ambas se requiere comparar los valores de la celda de la columna 2 o 3 (según sea el caso) con el valor de la celda de la columna 1, el cual ya debe estar validado al cumplirse la regla 1. De manera análoga se pueden obtener las siguientes:

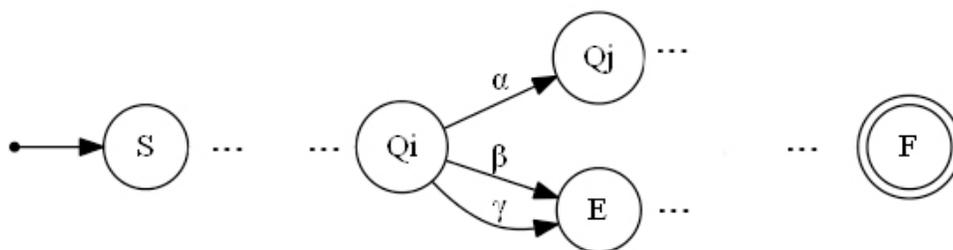
- Para validar la regla 4 antes falta validar las reglas 2 y 3.
- Para validar la regla 5 y la regla 7 antes falta validar la regla 4.

- Para validar la regla 6 y la regla 9 antes falta validar la regla 5.
- Para validar la regla 8 y la regla 10 antes falta validar la regla 7.

Ahora, para la construcción del autómata de validación, se presenta la siguiente notación:

- $V(c)$ = La celda c está vacía.
- $EP(c)$ = El valor de la celda c es un número entero positivo.
- $CR(i)$ = Se cumple la regla i .
- Ω = Manda un mensaje de error, indicando el motivo. Pinta de color rojo aquellas celdas involucradas.
- Λ = Retira los mensajes de error, si es que hay. Si la fila está completamente validada, entonces pinta las celdas de verde.

La idea se muestra en la Figura 2.9



donde:

$$\alpha = \frac{\neg V(c_{j1}) \wedge \dots \wedge \neg V(c_{jn}) \wedge EP(c_{j1}) \dots \wedge EP(c_{jn}) \wedge CR(j)}{\Lambda}$$

$$\beta = \frac{\neg V(c_{j1}) \wedge \dots \wedge \neg V(c_{jn}) \wedge (\neg EP(c_{j1}) \vee \dots \vee \neg EP(c_{jn}) \vee \neg CR(j))}{\Omega}$$

$$\gamma = \frac{V(c_{j1}) \vee \dots \vee V(c_{jn})}{\Lambda}$$

Figura 2.9: Idea para autómata de validación

- Se utilizará un autómata finito no-determinista traductor.
- S es un estado inicial, F un estado final y E un estado de error.

- Los operadores lógicos \neg, \vee, \wedge indican negación, disyunción y conjunción, respectivamente
- α, β y γ son transiciones del autómata. Al leer un elemento de entrada se transfieren de estado y al mismo tiempo entregan un elemento de salida.
- Un estado Q_k indica que la regla k ha sido validada. Se pretende indicar cómo es que se lleva a cabo esta validación.
- c_{j1}, \dots, c_{jn} son las celdas involucradas en la validación de la regla j .
- Se pone como ejemplo el estado Q_j . Para llegar a Q_j , desde Q_i , hace falta pasar por la transición α .
- La transición α indica que si la celda de la columna j no es vacía, su valor es un número entero positivo y que se cumple la regla j , entonces se dispara la función Λ y luego se avanza al estado Q_j . En otras palabras, el usuario escribió un valor válido que cumple la regla j ; luego, el sistema limpia posibles mensajes de error anteriores y pinta de color verde las celdas de la fila.
- La transición β indica que si la celda de la columna j no es vacía pero su valor no es un entero positivo o no se cumple la regla j , entonces se dispara la función Ω y se avanza al estado E . En otras palabras, el usuario ha escrito un valor que no es válido y no se cumple la regla j ; luego, el sistema manda cuál fue el error y pinta de color rojo las celdas involucradas.
- La transición γ indica que si la celda de la columna j es vacía, entonces se dispara la función Ω y se avanza al estado E . En otras palabras, el usuario no ha escrito algún valor que permita continuar con la validación de la regla j . No indica precisamente un error, así que el sistema limpia posibles mensajes de error anteriores.

Integrando los estados para todas las reglas, tomando en cuenta la dependencia entre validaciones, y sustituyendo el estado de error por el estado inicial, el autómata queda como en la Figura 2.10.

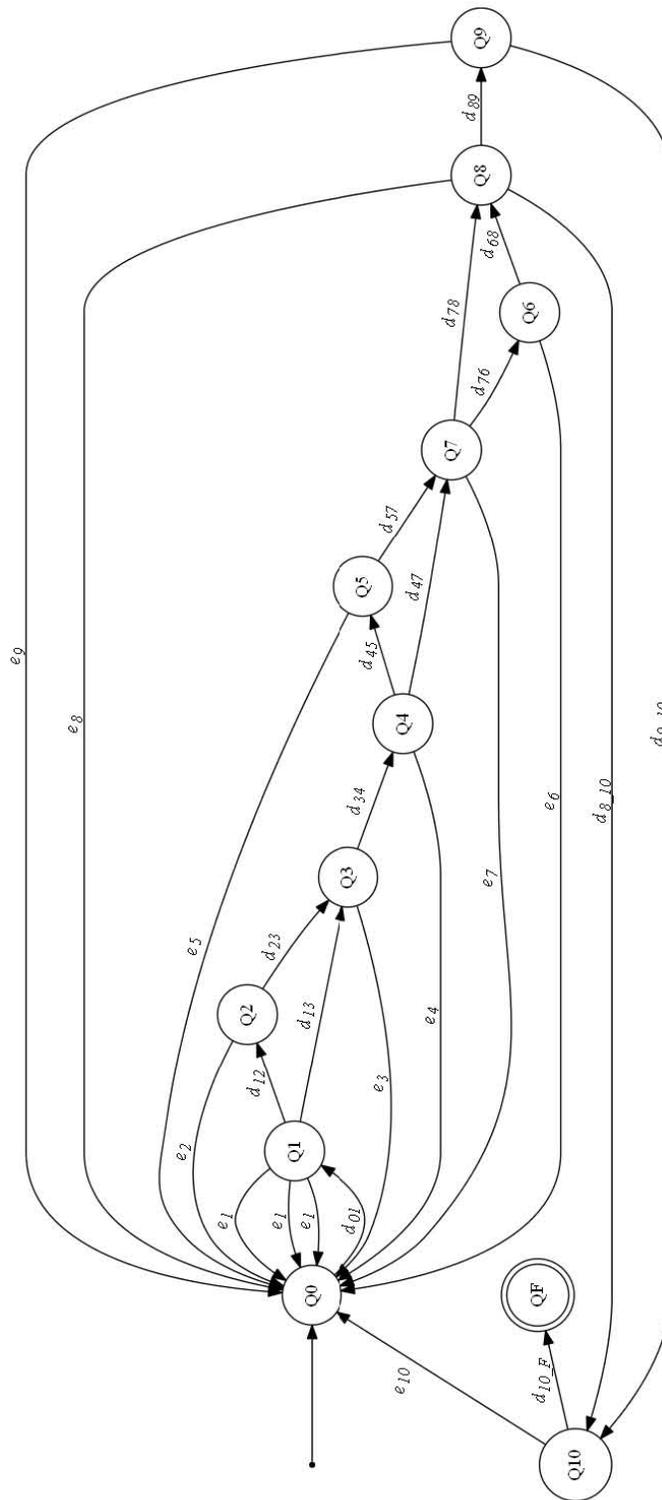


Figura 2.10: Autómata final para validación

Transiciones de la Figura 2.10:

- $\delta_{01} = \frac{\neg V(c_1) \wedge EP(c_1) \wedge CR(1)}{\Lambda}$
- $\delta_{12} = \frac{\neg V(c_2) \wedge EP(c_2) \wedge CR(2)}{\Lambda}$
- $\delta_{13} = \frac{\neg V(c_3) \wedge EP(c_3) \wedge CR(3)}{\Lambda}$
- $\delta_{23} = \delta_{13}$
- $\delta_{34} = \frac{CR(4)}{\Lambda}$
- $\delta_{45} = \frac{\neg V(c_6) \wedge EP(c_6) \wedge CR(5)}{\Lambda}$
- $\delta_{47} = \frac{\neg V(c_7) \wedge EP(c_7) \wedge CR(7)}{\Lambda}$
- $\delta_{57} = \delta_{47}$
- $\delta_{76} = \frac{\neg V(c_4) \wedge EP(c_4) \wedge CR(6)}{\Lambda}$
- $\delta_{78} = \frac{\neg V(c_5) \wedge EP(c_5) \wedge CR(8)}{\Lambda}$
- $\delta_{68} = \delta_{78}$
- $\delta_{89} = \frac{CR(9)}{\Lambda}$
- $\delta_{8.10} = \frac{CR(10)}{\Lambda}$
- $\delta_{9.10} = \delta_{89}$
- $\delta_{10-F} = \frac{CR(1) \wedge CR(2) \wedge CR(3) \wedge CR(4) \wedge CR(5) \wedge CR(6) \wedge CR(7) \wedge CR(8) \wedge CR(9) \wedge CR(10)}{\Lambda}$

Las transiciones e_i son análogas a la transición β , descrita anteriormente.

Diseño

El cuestionario consta de presentarle al usuario de Información Básica un formulario que le permita capturar sus datos acerca de las actividades de un organismo de alojamiento. Ver pantallas finales en las Figuras B.1 y B.2 del Apéndice B.

A continuación se muestra el caso de uso detallado para la carga de datos en el sistema, por parte del área de Información Básica.

Caso de uso: Carga de Información Básica.

Actor: Usuario Local, Administrador

Descripción: Cargar en el sistema los datos del área de Información Básica, por medio de un cuestionario con validaciones en *tiempo real*.

Pre-condición: El actor se ha autenticado.

Post-condición: El actor ha cargado datos acerca del organismo para el período solicitado.

Tabla 2.5: Caso de uso Carga de Información Básica

Actor	Sistema	Excepciones
1. Entra a la aplicación	2. Muestra el menú, indicando los módulos permitidos al usuario	
3. Selecciona la opción <i>Carga de Información Básica</i> .	4. En caso de usuario Administrador, el sistema mostrará una lista de organismos disponibles para captura; mientras que para el usuario local ya tendrá asignado un organismo y deberá pasar directamente al paso 6.	
5. Selecciona un organismo	6. Muestra una pantalla con tres opciones: <ul style="list-style-type: none"> • Carga de datos mes actual. • Carga de datos mes anterior. • Carga de datos otra fecha (sólamente Administrador). 	
7. Elige la opción <i>Carga de datos mes actual</i>	8. Muestra un cuestionario de captura de datos del primer día del mes al día actual. Continúa en el paso 15	
9. Elige la opción <i>Carga de datos mes anterior</i>	10. Muestra un cuestionario de captura de datos de todos los días del mes anterior al actual. Continúa en el paso 15	

<p>11. Elige la opción <i>Carga de datos otra fecha</i></p>	<p>12. Muestra:</p> <ul style="list-style-type: none"> ● campo de selección de mes y año (la fecha máxima a seleccionar es la actual, la mínima es Enero de 2000). ● botón Aceptar. ● botón Cancelar. 	
<p>13. Selecciona una fecha y da click en <i>Aceptar</i></p>	<p>14. Muestra un cuestionario de captura de datos de todos los días del mes y año seleccionado. Continúa en el paso 15</p>	
	<p>15. Muestra:</p> <ul style="list-style-type: none"> ● datos generales del organismo. ● tabla de captura de datos. ● botón Guardar. ● botón Limpiar. ● botón Cancelar. <p>El cuestionario es una tabla que tiene filas que representan días y siete columnas que representan las actividades que desarrolló el organismo en ese día. (Aquí es donde se debía aplicar la validación de reglas).</p>	
<p>16. Captura datos inválidos y cambia de campo de captura o da click en <i>Guardar</i></p>	<p>17. Indica con un mensaje el motivo de los datos rechazados y pinta con rojo las celdas del cuestionario involucradas. No se muestran todos los mensajes al mismo tiempo, pues esto genera demasiado <i>ruido</i> y genera confusión en el usuario; sólo se muestra un mensaje a la vez y seguirá marcando sólo ese mientras el usuario no lo corrija.</p>	

18. Captura datos válidos y cambia de campo de captura o da click en <i>Guardar</i>	19. Elimina los posibles mensaje de error anteriores y pinta de blanco(color por defecto) las celdas válidas. En caso que el renglón entero contenga datos válidos, todas las celdas de ese renglón se pintarán de verde. Además la barra de botones cambia a la siguiente: <ul style="list-style-type: none"> • botón Guardar y Continuar. • botón Editar. • botón Cancelar. 	
20. El usuario da click en <i>Guardar y Continuar</i>	21. Guarda los datos capturados y cambia a una pantalla de “Resumen” donde indica al usuario el estatus de la operación realizada	E1

Tabla 2.6: Excepciones Caso de uso Carga de Información Básica

ID	Descripción
E1	El sistema indica que ocurrió un problema interno en el sistema y que los datos no han sido guardados. Se indican detalles acerca del error.

Para el algoritmo de validación, la idea es la siguiente:

```

Input: tablaCuestionario
Output: true, false
begin
  esValido  $\leftarrow$  true
  for fila in tablaCuestionario do
    if  $\neg$  esVacía(fila) then
      | completarColumnasVacías(fila)
    end
  end
  foreach fila in tablaCuestionario do
    if  $\neg$  esVacía(fila) then
      | esValido  $\leftarrow$  validaDía(fila)
      | if  $\neg$  esValido then
        | | return false
      | else
        | | cambiarColorRenglon(verde, fila)
      | end
    end
  end
  return esValido
end

```

Algoritmo 7: *validaCuestionario*

Lo que se tiene es una función *validaCuestionario*, que se va a ejecutar cada que el usuario dé click en el botón *Guardar*. Recibe el arreglo de celdas de la tabla cuestionario a llenar, y devuelve un valor booleano; *true* para indicar que se puede proseguir a guardar el cuestionario en el sistema, y *false* para indicar que aún no se puede guardar.

Lo que hace es recorrer las filas que tienen al menos una celda capturada y rellena con ceros aquellas columnas de esa misma fila que se quedaron en “blanco”. Esto es para tener valores con que referenciarse a la hora de validar las diez reglas. Luego, de igual manera, el algoritmo vuelve a recorrer las filas y valida aquellas filas que tengan datos capturados. El algoritmo va pintando de verde aquellas filas que ya han sido validadas y se detiene cuando ha terminado de validar todas las filas o cuando encuentra el primer error para determinada fila.

A continuación se presenta la función *esNumeroEnteroPositivo* y *validaCelda*. Ésta última, encargada de dar a la captura de datos la funcionalidad de validación *al*

vuelo.

```

Input: celda
Output: true, false
begin
  if  $\neg$  esVacía(celda) then
    foreach char in getValor(celda) do
      if  $\neg$  esDigitoDel0al9(char) then
        mensajeError  $\leftarrow$  "Escriba sólo números enteros"
        mandarMensaje(Error, 10seg, mensajeError)
        cambiarColorCelda(rojo, celda)
        enfocarCelda(celda)
        return false
      end
    end
    cambiarColorCelda( blanco, celda)
    return true
  else
    return false
  end
end

```

Algoritmo 8: *esNumeroEnteroPositivo*

La función *esNumeroEnteroPositivo* recibe como parámetro una celda y devuelve un valor booleano; *true* para indicar si el valor que contiene la celda se trata de un número entero positivo, y *false* en caso contrario.

```

Input: celda
begin
  if esNumeroEnteroPositivo(celda) then
    fila  $\leftarrow$  getRenglonPadre(celda)
    validaDia(fila)
  end
end

```

Algoritmo 9: *validaCelda*

Lo que se tiene es una función *validaCelda*, que se va a ejecutar cada que el usuario termine de responder una celda y cambie a otra; si el valor de la celda es un número entero positivo, entonces el renglón de la celda afectada se pasará como parámetro y se procederá a hacer la validación (algunas celdas del mismo renglón pueden quedar

en blanco si el usuario no las ha capturado).

```

Input: renglonDia
Output: true, false
begin
  numDia  $\leftarrow$  getNumeroDeRenglon(renglonDia)
   $\sigma \leftarrow$  getDato1(organismo)
   $\kappa \leftarrow$  8
  c1  $\leftarrow$  getColumna1(renglonDia)
  c2  $\leftarrow$  getColumna2(renglonDia)
  c3  $\leftarrow$  getColumna3(renglonDia)
  c4  $\leftarrow$  getColumna4(renglonDia)
  c5  $\leftarrow$  getColumna5(renglonDia)
  c6  $\leftarrow$  getColumna6(renglonDia)
  c7  $\leftarrow$  getColumna7(renglonDia)
  renglonDiaAnterior  $\leftarrow$  getRenglonAnterior(renglonDia)
  c6*  $\leftarrow$  getColumna6(renglonDiaAnterior)
  c7*  $\leftarrow$  getColumna7(renglonDiaAnterior)
  if  $\neg$ esVacía(c1)  $\wedge$   $\neg$ validaRegla1( $\sigma$ , c1) then
    | return false
  end
  if  $\neg$ esVacía(c1)  $\wedge$   $\neg$ esVacía(c2)  $\wedge$   $\neg$ validaRegla2(c1, c2) then
    | return false
  end
  if  $\neg$ esVacía(c1)  $\wedge$   $\neg$ esVacía(c3)  $\wedge$   $\neg$ validaRegla3(c1, c3) then
    | return false
  end
  if  $\neg$ esVacía(c1)  $\wedge$   $\neg$ esVacía(c2)  $\wedge$   $\neg$ esVacía(c3)  $\wedge$   $\neg$ validaRegla4(c1, c2, c3) then
    | return false
  end
  if  $\neg$ esVacía(c2)  $\wedge$   $\neg$ esVacía(c6)  $\wedge$   $\neg$ validaRegla5(c2, c6) then
    | return false
  end
  if  $\neg$ esVacía(c3)  $\wedge$   $\neg$ esVacía(c7)  $\wedge$   $\neg$ validaRegla7(c3, c7) then
    | return false
  end
  if  $\neg$ esVacía(c4)  $\wedge$   $\neg$ esVacía(c6)  $\wedge$   $\neg$ validaRegla6(c4, c6) then
    | return false
  end
  if  $\neg$ esVacía(c5)  $\wedge$   $\neg$ esVacía(c7)  $\wedge$   $\neg$ validaRegla8(c5, c7) then
    | return false
  end
  if  $\neg$ esVacía(c4)  $\wedge$   $\neg$ esVacía(c6)  $\wedge$   $\neg$ validaRegla9(c4, c6, c6*) then
    | return false
  end
  if  $\neg$ esVacía(c5)  $\wedge$   $\neg$ esVacía(c7)  $\wedge$   $\neg$ validaRegla10(c5, c7, c7*) then
    | return false
  end
  if  $\neg$ esVacía(c1)  $\wedge$   $\neg$ esVacía(c2)  $\wedge$   $\neg$ esVacía(c3)  $\wedge$   $\neg$ esVacía(c4)  $\wedge$   $\neg$ esVacía(c5)  $\wedge$   $\neg$ esVacía(c6)  $\wedge$ 
   $\neg$ esVacía(c7) then
    | return true
  end
  return true
end

```

Algoritmo 10: validaDia

Lo que se tiene es una función *validaDia*, que se va a ejecutar cada que es llamada desde las funciones *validaCuestionario* y *validaCelda*.

La función recibe un renglón de la tabla de cuestionario, primero establece todos

aquellos valores que se van a utilizar para la validación de las reglas, y luego va chequeando que se cumplan estas reglas una por una.

A partir del renglón pasado como parámetro, se obtiene el número de día que representa ese renglón y siete celdas correspondientes a las siete columnas del cuestionario. Además se obtiene una referencia al renglón anterior y se obtienen las celdas de las columnas 6 y 7. Por último, se tienen los valores σ y κ (σ : un valor atributo del organismo y κ : un valor de una constante). A partir de todos ellos se verificará que el renglón cumpla con las diez reglas.

Algunas celdas del renglón puede que hayan quedado en “blanco”, sin embargo, lo que se hace es seguir el diseño del autómata, donde indica que si las celdas responsables de la validación de una regla aún no han sido capturadas, entonces no se puede proseguir a la validación de la misma.

A continuación cada una de las diez validaciones. La función *mandarMensaje(tipo, tiempo, texto)* se refiere a mandar un mensaje emergente al usuario que indique un *texto* en particular, dure un *tiempo* determinado en la pantalla, y que dependiendo el *tipo* de mensaje se muestre de diferente color. La función *cambiarColorCelda(color, celda)* le asigna un *color* determinado a la *celda* pasada por parámetro. Y por último, la función *enfocarCelda(celda)* hace que el cursor del usuario apunte hacia la *celda* pasada por parámetro.

```

begin
  if esNumeroEnteroPositivo( $c_1$ )  $\wedge$   $c_1 > \sigma$  then
    mensajeError  $\leftarrow$  “En el día ” + numDia + “, corregir
      el número de cuartos disponibles, pues debe ser menor
      o igual al número de cuartos registrados”
    mandarMensaje(Error, 10seg, mensajeError)
    cambiarColorCelda(rojo,  $c_1$ )
    enfocarCelda( $c_1$ )
    return false
  else
    cambiarColorCelda(blanco,  $c_1$ )
    return true
  end
end

```

Algoritmo 11: validaRegla1

```

begin
  if esNumeroEnteroPositivo( $c_2$ )  $\wedge$   $c_1 < c_2$  then
    mensajeError  $\leftarrow$  "En el día " + numDia + ", corregir
      el número de cuartos ocupados residentes, pues debe ser menor
      o igual al número de cuartos disponibles"
    mandarMensaje(Error, 10seg, mensajeError)
    cambiarColorCelda(rojo,  $c_1$ )
    cambiarColorCelda(rojo,  $c_2$ )
    enfocarCelda( $c_2$ )
    return false
  else
    cambiarColorCelda(blanco,  $c_1$ )
    cambiarColorCelda(blanco,  $c_2$ )
    return true
  end
end

```

Algoritmo 12: validaRegla2

```

begin
  if esNumeroEnteroPositivo( $c_3$ )  $\wedge$   $c_1 < c_3$  then
    mensajeError  $\leftarrow$  "En el día " + numDia + ", corregir
      el número de cuartos ocupados no residentes, pues debe ser
      menor o igual al número de cuartos disponibles"
    mandarMensaje(Error, 10seg, mensajeError)
    cambiarColorCelda(rojo,  $c_1$ )
    cambiarColorCelda(rojo,  $c_3$ )
    enfocarCelda( $c_3$ )
    return false
  else
    cambiarColorCelda(blanco,  $c_1$ )
    cambiarColorCelda(blanco,  $c_3$ )
    return true
  end
end

```

Algoritmo 13: validaRegla3

```

begin
  if  $c_1 < c_2 + c_3$  then
    mensajeError  $\leftarrow$  “En el día ” + numDia + “, corregir
      el número total de cuartos ocupados, pues debe ser menor
      o igual al número de cuartos disponibles”
    mandarMensaje(Error, 10seg, mensajeError)
    cambiarColorCelda(rojo,  $c_1$ )
    cambiarColorCelda(rojo,  $c_2$ )
    cambiarColorCelda(rojo,  $c_3$ )
    enfocarCelda( $c_3$ )
    return false
  else
    cambiarColorCelda( blanco,  $c_1$ )
    cambiarColorCelda( blanco,  $c_2$ )
    cambiarColorCelda( blanco,  $c_3$ )
    return true
  end
end
end

```

Algoritmo 14: validaRegla4

```

begin
  if esNumeroEnteroPositivo( $c_6$ )  $\wedge$   $c_6 > c_2 * \kappa$  then
    mensajeError  $\leftarrow$  “En el día ” + numDia + “, corregir
      el número de turistas noche residentes, pues debe ser
      menor o igual al número de cuartos
      residentes por el factor de ocupación”
    mandarMensaje(Error, 10seg, mensajeError)
    cambiarColorCelda(rojo,  $c_2$ )
    cambiarColorCelda(rojo,  $c_6$ )
    enfocarCelda( $c_6$ )
    return false
  else
    cambiarColorCelda( blanco,  $c_2$ )
    cambiarColorCelda( blanco,  $c_6$ )
    return true
  end
end
end

```

Algoritmo 15: validaRegla5

```

begin
  if esNumeroEnteroPositivo( $c_7$ )  $\wedge$   $c_7 > c_3 * \kappa$  then
    mensajeError  $\leftarrow$  “En el día ” + numDia + “, corregir
      el número de turistas noche no residentes, pues debe ser
      menor o igual al número de cuartos
      residentes por el factor de ocupación”
    mandarMensaje(Error, 10seg, mensajeError)
    cambiarColorCelda(rojo,  $c_3$ )
    cambiarColorCelda(rojo,  $c_7$ )
    enfocarCelda( $c_7$ )
    return false
  else
    cambiarColorCelda(blanco,  $c_3$ )
    cambiarColorCelda(blanco,  $c_7$ )
    return true
  end
end

```

Algoritmo 16: validaRegla7

```

begin
  if esNumeroEnteroPositivo( $c_4$ )  $\wedge$   $c_6 > c_4$  then
    mensajeError  $\leftarrow$  “En el día ” + numDia + “, corregir el
      número de turistas noche residentes, pues debe ser menor
      o igual al número de llegada de turistas residentes”
    mandarMensaje(Error, 10seg, mensajeError)
    cambiarColorCelda(rojo,  $c_4$ )
    cambiarColorCelda(rojo,  $c_6$ )
    enfocarCelda( $c_6$ )
    return false
  else
    cambiarColorCelda(blanco,  $c_4$ )
    cambiarColorCelda(blanco,  $c_6$ )
    return true
  end
end

```

Algoritmo 17: validaRegla6

```

begin
  if esNumeroEnteroPositivo( $c_5$ )  $\wedge$   $c_7 > c_5$  then
    mensajeError  $\leftarrow$  "En el día " + numDia + ", corregir el
      número de turistas noche no residentes, pues debe ser menor
      o igual al número de cuartos de llegada de turistas no residentes"
    mandarMensaje(Error, 10seg, mensajeError)
    cambiarColorCelda(rojo,  $c_5$ )
    cambiarColorCelda(rojo,  $c_7$ )
    enfocarCelda( $c_7$ )
    return false
  else
    cambiarColorCelda(blanco,  $c_5$ )
    cambiarColorCelda(blanco,  $c_7$ )
    return true
  end
end

```

Algoritmo 18: validaRegla8

```

begin
  if  $c_6 > c_4 + c_6^*$  then
    mensajeError  $\leftarrow$  "En el día " + numDia + ", corregir el
      número de turistas noche residentes, pues debe ser
      menor o igual al número de llegada de turistas residentes
      más el número de turistas noche residentes del día anterior"
    mandarMensaje(Error, 10seg, mensajeError)
    cambiarColorCelda(rojo,  $c_4$ )
    cambiarColorCelda(rojo,  $c_6$ )
    cambiarColorCelda(rojo,  $c_6^*$ )
    enfocarCelda( $c_6$ )
    return false
  else
    cambiarColorCelda(blanco,  $c_4$ )
    cambiarColorCelda(blanco,  $c_6$ )
    cambiarColorCelda(blanco,  $c_6^*$ )
    return true
  end
end

```

Algoritmo 19: validaRegla9

```

begin
  if  $c_7 > c_5 + c_7^*$  then
    mensajeError  $\leftarrow$  “En el día ” + numDia + “, corregir el
      número de turistas noche no residentes, pues debe ser
      menor o igual al número de llegada de turistas no residentes
      más el número de turistas noche no residentes del día anterior”
    mandarMensaje(Error, 10seg, mensajeError)
    cambiarColorCelda(rojo,  $c_5$ )
    cambiarColorCelda(rojo,  $c_7$ )
    cambiarColorCelda(rojo,  $c_7^*$ )
    enfocarCelda( $c_7$ )
    return false
  else
    cambiarColorCelda( blanco,  $c_5$ )
    cambiarColorCelda( blanco,  $c_7$ )
    cambiarColorCelda( blanco,  $c_7^*$ )
    return true
  end
end
end

```

Algoritmo 20: validaRegla10

2.5 Diseño de la Administración de Catálogos y de Sistema

Debido a que ya se revisaron los componentes de la parte de *Captación de información*, que representaban el principal objetivo de este trabajo, ahora se presentan, de manera muy sintetizada, las secciones de *Administración del sistema*, *Catálogos generales*, *Monitoreo de organismos* y *Reportes*.

En los componentes de *Administración del sistema* y de *Catálogos generales* se requerían módulos de alta, baja, modificación y consulta. Se requería crear una interfaz que permitiese administrar las tablas catálogo de la base de datos y que contase con validaciones propias de cada campo, es decir, si un campo era de tipo:

- fecha, entonces debía mostrar un calendario que permitiese seleccionar una fecha; este componente, además de restringir y dar formato a la fecha, hace más cómoda la captura al usuario.
- numérico, entonces debía restringir máximos, mínimos y, en su caso, formatos;

por ejemplo, si se trataba de Código Postal debía permitir guardar números de cinco dígitos.

- *texto*, entonces debía restringir tamaño máximo, mínimo, formato o caracteres válidos; por ejemplo para la validación de correo electrónico se debía tomar en cuenta que el campo tuviera el arroba y que terminase con .com, .net, .org, .mx, etc.

Otro tipo de validaciones, debía ser no permitir duplicados en el alta de registros. Por ejemplo, en el caso de dar de alta un nuevo usuario, había que fijarse que el login no estuviera previamente registrado en la base de datos.

En cuanto a los módulos de seguridad que abarca la *Administración del sistema*, se debía identificar en cada acceso a la aplicación: el usuario, su rol, permisos asignados, y determinar si se debía mostrar el contenido al usuario o no. Para lograr esto, se debía tener apoyo de la base de datos, el menú de la aplicación debía mostrar sólo aquellos módulos permitidos para el rol de usuario con que se accedió al sistema; además de validar en cada página del sistema (por medio de la URL) el componente al que pertenecía y si el usuario debía tener acceso o no.

Por otro lado, el componente *Monitoreo de organismos de alojamiento* representó ser complejo, pues interactuaban bastantes reglas de negocio.

A continuación se presenta la estrategia utilizada para obtener dos tipos de Reportes del área de Información Básica, los cuales se pueden apreciar en las Figuras D.1 y D.2 del Apéndice D

- En la Figura D.1 del Apéndice D, se puede ver una tabla donde se tienen como filas a un grupo de organismos de alojamiento y como columnas un conjunto de fechas en particular, seleccionadas previamente. El contenido indica los días en que los organismos han capturado sus datos para el Área de Información Básica.

Para llevar a cabo este componente se tomaron en cuenta que los parámetros debían ser: un arreglo de grupos de organismos de alojamiento, una arreglo de clasificación de organismos de alojamiento, una actividad de organismos de alojamiento, año y mes (o semana) seleccionados previamente (por medio de la interfaz gráfica del sistema).

Luego, se realizó una consulta a la base de datos y por medio de consultas donde intervenían las tablas **Cuestionario**, **ClasificacionOrgAlojamiento**, **GruposOrgAlojamiento**, **Municipios**, **OrgAlojamientoHistorico**,

Estados y OrgAlojamientoHistoGruposOAlojamiento.

Primero se debía hacer una pre-selección de los organismos de alojamiento que cumplieran formar parte del grupo, clasificación y actividad para el periodo seleccionado(año, mes o semana).

Posteriormente, se hacia una combinación con la tabla del cuestionario por medio de la llave del organismo de alojamiento, se filtraban los resultados igualmente por período, y se le aplicaba la sentencia *PIVOT* para cambiar las filas de las fechas por columnas.

Finalmente, se le daba el formato a la salida para que se agruparan los organismos de acuerdo a su estado, municipio y clasificación.

- En la Figura D.2 del Apéndice D, podemos ver una tabla con información de las actividades de los organismos de alojamiento, reflejadas en 20 indicadores. Dichos datos se toman de un cierre de monitoreo para el Área de Información Básica. Es decir, se elige un grupo de organismos de alojamiento y se les aplica un cierre periódico en cuanto a los datos que se registran por medio del Cuestionario: Se toman en cuenta los datos de aquellos organismos que si lograron completar su cuestionario a tiempo, y a los que no, se les aplica un fórmula de expansión¹¹, para poder obtener una aproximación acerca de las actividades de todos los organismos de alojamiento del país.

Para llevar a cabo este componente se tomaron en cuenta que los parámetros debían ser: un arreglo de grupos de organismos de alojamiento, una arreglo de clasificación de organismos de alojamiento, una actividad de organismos de alojamiento, año y mes (o semana) seleccionados previamente (por medio de la interfaz gráfica del sistema).

Luego, se realizó una consulta a la base de datos y por medio de consultas donde intervenían las tablas **Cuestionario**, **ClasificacionOrgAlojamiento**, **GruposOrgAlojamiento**, **Municipios**, **OrgAlojamientoHistorico**, **Estados** y **OrgAlojamientoHistoGruposOAlojamiento**.

Primero se debía hacer una pre-selección de los organismos de alojamiento que cumplieran formar parte del grupo, clasificación y actividad para el periodo seleccionado(año, mes o semana).

Posteriormente, se hacia una combinación con la tabla del cuestionario por medio de la llave del organismo de alojamiento, se filtraban los resultados igualmente por período, y se obtenian cada uno de los 20 indicadores, a partir de los datos de las siete columnas del cuestionario. Ya que se tenían los datos

¹¹Dicha fórmula de expansión fue proporcionada por el cliente

de los organismos que sí completaron su cuestionario, entonces se buscaba aquellos que no tuviesen registros de cuestionario para ese período, se les aplicaba la fórmula de expansión a partir de datos de organismos similares y se sumaban ambas partes.

Finalmente, se le daba el formato a la salida para que se agruparan los datos de acuerdo a la actividad, grupo y clasificación de los organismos en forma de agregación y se guardaban los resultados en la tabla de cierres de monitoreo.

2.6 Arquitectura del Sistema

El sistema se construyó bajo la arquitectura Cliente-Servidor, por lo que, los usuarios se pueden conectar desde distintos puntos de la República Mexicana, a través de un link, acceso a Internet y un navegador web.

La implementación del sistema se llevó a cabo por medio de las siguientes tecnologías:

- Microsoft Visual C#
Este lenguaje de programación orientado a objetos se utilizó para tener la infraestructura y control lógico del sistema.
- Microsoft SQL Server
Se utilizó tanto para manejo como para análisis de la base de datos y el data warehouse de la aplicación.
 - OLTP
La base de datos se construyó en un modelo relacional. Esta herramienta atiende las tareas frecuentes de lectura y escritura de datos en el sistema.
 - OLAP
Se utilizaron los módulos de Microsoft SQL Server Analysis Services (SSAS) y Excel Power Pivot para funciones de procesamiento analítico en línea (OLAP) y de minería de datos.
- Microsoft ASPX .NET
Se utilizó para desarrollar la Interfaz Gráfica de Usuario. Por medio de *scripts* y, en conjunto con C#, atiende las peticiones del cliente, entregando código en HTML al navegador del usuario para que éste pueda visualizar la aplicación. Además, se utilizan tecnologías como CSS, Javascript y Ajax para presentar el sistema de una manera dinámica y limpia, pero sobretodo usable.

- Microsoft Internet Information Services (IIS)
Se utilizó como servidor web de la aplicación. Entre otros servicios, ofrece transferencia de datos por medio del protocolo HTTP/HTTPS, que es por el cual se comunicaran las máquinas cliente con el servidor.
- Microsoft SharePoint
Se utilizó como contenedor de la aplicación. Entre las principales funciones fue que ayudó para la integración de componentes, visualización de reportes, y para facilitar la sección de búsqueda dentro de la aplicación. Además, se agregan las siguientes páginas solicitadas por el cliente: Inicio, Mapa de Sitio, Glosario, Contacto, Documentos y Publicaciones.
- *Drivers* de conexión desde la aplicación con C# hacia MS SQL Server, MS Excel y MS Access para lectura y escritura de datos.

A continuación se presenta la Figura 2.11, la cual muestra la arquitectura final del sistema desarrollado. De parte del cliente, al servidor web le llegan peticiones que son atendidas por el IIS. El IIS, en conjunto con el framework de ASP .NET, dirige las peticiones al Sharepoint. Este último contiene la aplicación web del sistema, que podrá apoyarse del SQL Server y encargarse de devolver respuesta a la petición.

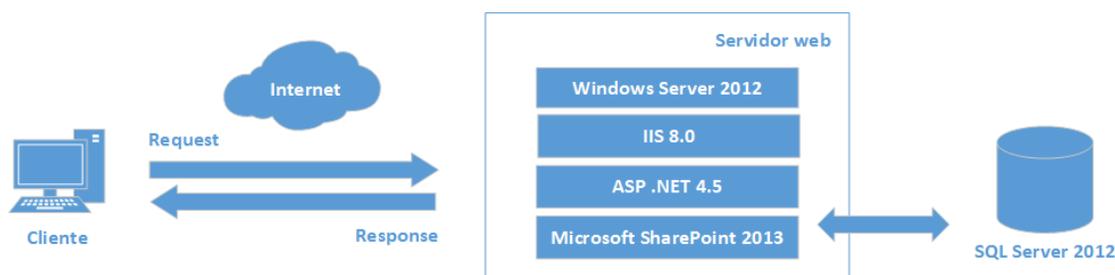


Figura 2.11: Arquitectura del sistema

Por otro lado, el patrón de diseño utilizado en el sistema fue el de Modelo-Vista-Controlador. La Figura 2.11 ilustra los diversos componentes de ASP.NET MVC, cómo se relacionan entre ellos, y el flujo en que se desarrollan las peticiones y las respuestas. ASP.NET MVC incorpora un motor de enrutamiento que se sitúa delante de los demás componentes MVC. El motor de enrutamiento recibe solicitudes HTTP entrantes y las encamina a un método de controlador. Posteriormente, el trabajo del controlador está en extraer información de la solicitud entrante e interactuar con la capa de modelo. La capa de modelo es la capa que lleva la lógica de negocio y se

comunica con la base de datos. Una vez que el controlador ha terminado de utilizar el modelo, crea una vista a la cual le proporciona los datos del modelo para usarla en la respuesta de salida.

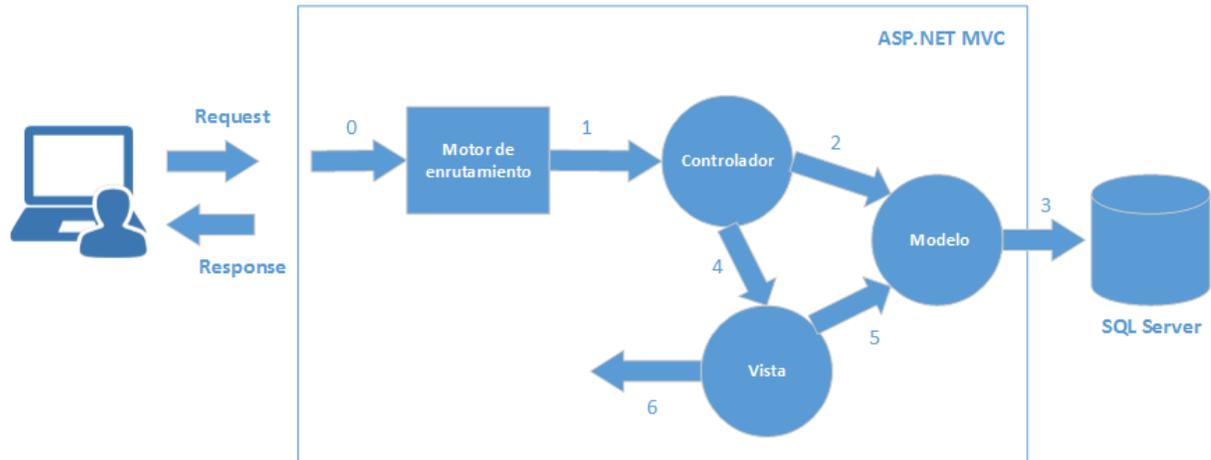


Figura 2.12: Arquitectura de ASP.NET MVC

Capítulo 3

Resultados

El sistema desarrollado, por parte de todo el equipo, es un conjunto de elementos orientados a la recolección, almacenamiento, procesamiento y explotación de datos, para su uso posterior por parte del cliente. En general, tiene entre sus principales características:

- Administración de Catálogos y de Sistema.
- Disponibilidad de información histórica de cambios.
- Integración de distintas fuentes de datos.
- Módulo específico de captura de datos con validación *al vuelo*.
- Información adecuada al perfil de cada usuario.
- Informes dinámicos, flexibles e interactivos.
- Trabaja por medio de un módulo de procesamiento analítico de datos en línea (OLAP).
- Rapidez en el tiempo de respuesta.

Mi participación en el sistema, fue llevar la responsabilidad de implementar las funcionalidades presentadas en este reporte. En concreto, fueron los módulos de: *Administración del sistema*, *Catálogos generales*, *Captación de información*, *Monitoreo de organismos de alojamiento* y la sección de *Reportes de Información Básica*. Sólomente no estuve presente en el desarrollo de los *Reportes de Información Externa* y la implementación del contenedor de la aplicación. Contribuí en el desarrollo tanto de la parte de OLAP como la de OLTP, pero no fui

asignado para el diseño del data warehouse ni la elaboración de reportes con OLAP.

Por otra parte, para llevar a cabo este desarrollo se requirió de un grupo de tres programadores (yo incluido) de tiempo completo. Los tiempos en que se había acordado la entrega del producto fue en un inicio de ocho meses, sin embargo, algunos reajustes en los requerimientos del sistema, por parte del cliente, llevaron a extender el plazo a los diez meses.

Para poder terminar el proyecto en ese tiempo, hubo que organizar al equipo de trabajo por medio de dos metodologías ágiles: se utilizó *scrum* durante los primeros cinco meses y *extreme programming* en los cinco posteriores. Esto fue porque al principio, se tenía una idea general acerca de lo que debía hacer el sistema. Es decir, se debía analizar cuáles eran los componentes del sistema predecesor que se debía emular, cuáles rehacer, cuáles deshechar y añadir los nuevos.

Se tuvo bastante trabajo y la metodología *scrum* nos ayudó a avanzar rápidamente con la implementación de la estructura del sistema. Durante ese tiempo todos los componentes a desarrollar empezaban de cero, se tenía una idea general acerca de lo que debía hacer el componente, sin embargo, no se conocían algunos detalles de las reglas de negocio que pudiesen aplicarse. La programación de los componentes fue iterativa e incremental; se definían las tareas para los sprints y se comenzaba a programar. Más adelante, ya se tenía la estructura del componente y había que implementar aquellos detalles faltantes que sólo el cliente conocía. Fue entonces que se cambió a *extreme programming*: se trabajaba a diario con una persona, encargada por el cliente, en una sesión de una hora a dos, para especificar los detalles faltantes y al mismo tiempo hacer pruebas del sistema. Esto nos dejaba nuevas tareas para los programadores al final de la sesión; tareas de las cuales se debía entregar un avance para la sesión siguiente. Los resultados fueron bastante satisfactorios; en la recta final del proyecto sólo quedaba generar documentación acerca del sistema desarrollado y poner a disposición del cliente el producto.

A fecha en que se escribió este reporte, el sistema ha sido liberado exitosamente y ya se ha comenzado con las etapas de capacitación para los usuarios, así como la de mantenimiento del sistema. Con la colaboración de nuestro equipo de desarrollo, la liberación de la aplicación fue llevada a cabo por el área de Soporte Técnico del cliente, y la puesta en marcha en los servidores estuvo a cargo de su área de Redes de Cómputo.

Capítulo 4

Conclusiones y recomendaciones

A lo largo de los capítulos 1, 2 y 3 se presentó la problemática a resolver, el análisis detallado, las herramientas utilizadas y los resultados obtenidos del sistema desarrollado. La implementación de sistemas de información que se apoyan de minería de datos¹ para el soporte de decisiones es una tarea muy especializada, requiere de un entendimiento total de las reglas del negocio y una idea muy específica para saber qué se pretende de ese sistema. Afortunadamente, nuestro cliente lleva varios años de experiencia en su materia y nos apoyó (al equipo de desarrollo) con detalladas especificaciones acerca de los resultados que debía entregar el sistema.

La experiencia obtenida en la construcción de este sistema fue el producto de aplicar los conocimientos y bases adquiridas en la carrera. El uso de la lógica matemática para resolver problemas, la abstracción de modelos para construir los componentes del sistema y la organización que se requiere dentro del grupo de desarrollo fueron las principales herramientas que me permitieron aportar con mi granito de arena para el éxito de este proyecto.

En relación al trabajo presentado y a los procesos de extracción, transformación y carga de datos en sistemas de información, se puede concluir que: son procesos automatizados que ayudan al usuario a no tener que hacer tareas repetitivas y agobiantes como completar datos faltantes, identificar valores ruidosos, incompletos o fuera de lugar, y reemplazarlos por datos válidos, una y otra vez, archivo por archivo, período tras período.

¹En este caso se trató de un sistema que se apoya de un módulo de procesamiento analítico de datos en línea (OLAP, por sus siglas en inglés)

Por otra parte, el cuestionario con validaciones *al vuelo* fue de gran ayuda, pues el usuario al no saber al *pie de la letra* todo lo que se debe validar, le es más fácil identificar que los datos capturados son erróneos, por medio de colores y mensajes oportunos, y luego poderlos corregir de manera más eficaz si se le indica exactamente cuál fue y dónde estuvo el problema. Un sistema bien diseñado con una buena retroalimentación de mensajes le dejan una buena experiencia al usuario, le permiten llevar a cabo sus tareas de forma fácil y rápida, pero sobre todo le permiten cumplir sus necesidades.

En cuanto al desarrollo de los módulos de administración de catálogos y de sistema, se puede concluir que: son componentes básicos pero de gran importancia en una aplicación, su correcto funcionamiento permite al usuario tener un control y un dominio acerca de las entidades que participan dentro de su modelo de BD. Su mal funcionamiento hace que el usuario busque otros medios para llevar a cabo esas mismas tareas; puede hacer que usuarios no-expertos busquen *scripts* en SQL y administren (de manera muy peligrosa) los datos directamente, tal y como sucedió con el sistema predecesor que tenía el cliente.

Además, aunque a mí no me tocó desarrollar la parte de minería de datos, sí estuve presente en la de generación de algunos reportes y en la fase de pruebas con el usuario, y pude observar su trabajo del día a día y los períodos en que debía utilizar el sistema para la generación de reportes y la importancia que éstos representan para él. Después de ingresar múltiples registros de datos, el usuario al fin puede obtener de manera precisa y entendible información acerca de lo que le *dicen* sus datos, información útil que puede llevarle a dar un giro positivo a las actividades dentro del *negocio*. Me refiero a que, tenemos a la minería de datos como una potente herramienta para el análisis de grandes volúmenes de datos, y es a través de ella que podemos obtener información novedosa y útil sobre los datos ya conocidos; los resultados se presentan por medio de reportes, visualmente pueden representarse por medio de gráficas y vienen acompañados de cifras o valores; sin embargo, su interpretación humana es la que representa realmente la verdadera importancia para la toma de decisiones.

Por último, algunos de los aspectos que podrían mejorarse en la implementación actual son:

- Recuperación de la carga de datos en Información Externa
El sistema actual no asegura que durante los procesos de ETL ocurra un fallo

en el sistema destino y confirme que los datos han sido guardados. Es decir, una vez extraído el archivo, el sistema procesa los datos para transformación y carga, y mientras lo hace es posible que haya alguna falla en el sistema (como pérdida de energía eléctrica) y no se complete el proceso. El usuario tendría que cargar de nuevo su archivo, pues la configuración actual no guardaría los cambios, pues utiliza ROLL BACK y éste deshace cualquier cambio si es que no se concluyó el proceso. Una posible solución sería que el sistema identifique cuándo fue que sucedió una falla y aliste aquellos procesos de ETL que se quedaron en la fase Transformación y Carga, por medio de la bitácora de cambios. Luego, cargar de nuevo el archivo desde su carpeta de “home/archivosCargados/”, llevar a cabo de nuevo el proceso y, finalmente, dar aviso al usuario.

- Ejecución en varios hilos para la carga de datos en Información Externa
La ejecución concurrente podría utilizarse para la carga de fuentes. Los archivos a cargar traen datos por medio grandes cantidades de filas y un número determinado de columnas. Las filas son independientes entre sí, por lo que bien se podría repartir el procesamiento de estos datos por medio de un número determinado de hilos y, así, agilizar la operación.

Apéndice A

Carga de Información Externa

🏠 Cargado de archivos al servidor Jueves 20 mar. 2014 - Alan Gutiérrez Salr

Cargado de archivos al servidor

Fuente 1 Fuente 2 Fuente 3 Fuente 4 Fuente 5 Fuente 6

Fuente 2

Elegir archivo a procesar, fuente 2. Viajeros Inter... bril 2013.xlsx

Figura A.1: Carga de Información Externa

Apéndice B

Carga de Información Básica

Error

En el día 1, corregir el número de Turistas Noche Residentes (321), pues debe ser menor o igual al número de Cuartos Ocupados Residentes con un Factor de Ocupación por Cuartos de 8 (0*8=0).

Datos Generales

Nombre del establecimiento: HOTEL MISION AGUASCALIENTES ZONA SUR

Categoría: 4 estrellas Año: 2014

No. total de cuartos: 108 Mes: Marzo

Entidad Federativa: Aguascalientes Localidad: AGUASCALIENTES

Questionario

Día	Cuartos Disponibles	Cuartos ocupados (A)		Llegada de turistas (B)		No. de Turistas Noche (C)	
		Residentes	No Residentes	Residentes	No Residentes	Residentes	No Residentes
1	24	0	12	23	87	321	
2							
3							
4							
5							
6							
7							
8							
9							
10							

Figura B.1: Carga de Información Básica, validando la regla 6 en el día 1

Datos Generales		Nombre del establecimiento		Categoría		No. total de cuartos		Entidad Federativa		Año		Mes		Localidad	
		HOTEL MISION AGUASCALIENTES ZONA SUR		4 estrellas		108		Aguascalientes		2014		Marzo		AGUASCALIENTES	

Error															
Los Turistas Noche Residentes del día 2 (272), deben ser menor o igual que los Turistas Noche Residentes día anterior (34), mas Llegada de Turistas Residentes del día 2 (34). ES decir 272 <= 68 (34+34).															
Cuestionario		Cuartos ocupados (A)		Llegada de turistas (B)		No. de Turistas Noche (C)									
Día	Cuartos Disponibles	Residentes	No Residentes	Residentes	No Residentes	Residentes	No Residentes								
1	108	12	12	23	67	34	67								
2	108	35	35	34	35	272	35								
3															
4															
5															
6															
7															
8															
9															
10															

Figura B.2: Carga de Información Básica, validando la regla 9 en el día 2

Apéndice C

Administración de Catálogos del Sistema

The screenshot shows a web interface for adding a new federal entity. At the top, there is a breadcrumb trail: Inicio > Catálogo > Datos de la Entidad Federativa. The date and user information 'Jueves 20 mar. 2014 - Alan Gutiérrez Salir' are visible in the top right. The main heading is 'Nueva Entidad Federativa'. Below it, the section 'Datos Generales' contains three fields: 'Pais' with the value 'MEXICO' and a red error message 'Este campo es obligatorio'; 'Nombre de la Entidad' with a placeholder 'Capture el nombre de la entidad'; and 'Habilitado' with radio buttons for 'Si' (selected) and 'No'. At the bottom right, there are two buttons: 'Agregar' (blue) and 'Cancelar' (white).

Figura C.1: Ejemplo Catálogo - Alta

Inicio > Catálogo de Entidades Federativas Jueves 20 mar. 2014 - Alan Gutiérrez Salm

Catálogo de entidades federativas

Estado	Habilitado	Habilitar/Deshabilitar	Consultar
Aguascalientes	Si	Deshabilitar	Seleccionar
Baja California	Si	Deshabilitar	Seleccionar
Baja California Sur	Si	Deshabilitar	Seleccionar
Campeche	Si	Deshabilitar	Seleccionar
Chiapas	Si	Deshabilitar	Seleccionar
Chihuahua	Si	Deshabilitar	Seleccionar
Coahuila	Si	Deshabilitar	Seleccionar
Colima	Si	Deshabilitar	Seleccionar
Distrito Federal	Si	Deshabilitar	Seleccionar
Durango	Si	Deshabilitar	Seleccionar

<< < 1 2 3 4 > >> Ir a página: Registros por página: Registros: 1 a 10 de 34

Figura C.2: Ejemplo Catálogo - Listado y Búsqueda

Inicio > Catálogo > Datos de la Entidad Federativa Jueves 20 mar. 2014 - Alan Gutiérrez Salm

Consulta de la Entidad Federativa

Datos Generales

Pais	MEXICO
Nombre de la Entidad	Coahuila
Habilitado	Si

Figura C.3: Ejemplo Catálogo - Consulta

Inicio > Catálogo > Datos de la Entidad Federativa Jueves 20 mar. 2014 - Alan Gutiérrez Salm

Edición de la Entidad Federativa

Datos Generales

Pais	<input type="text" value="MEXICO"/>
Nombre de la Entidad	<input type="text" value="Coahuila"/>
Habilitado	<input checked="" type="radio"/> SI <input type="radio"/> No

Figura C.4: Ejemplo Catálogo - Modificación

Inicio > Catálogo de Entidades Federativas

Catálogo de entidades federativas

Aviso
 Se deshabilitó el estado Baja California Sur correctamente

▼ Filtrar resultados
⊞ Agregar Nuevo

Estado	Habilitado	Habilitar/Deshabilitar	Consultar
Aguascalientes	Si	Deshabilitar	Seleccionar
Baja California	Si	Deshabilitar	Seleccionar
Baja California Sur	No	Habilitar	Seleccionar
Campeche	Si	Deshabilitar	Seleccionar
Chiapas	Si	Deshabilitar	Seleccionar
Chihuahua	Si	Deshabilitar	Seleccionar
Coahuila	Si	Deshabilitar	Seleccionar
Colima	Si	Deshabilitar	Seleccionar
Distrito Federal	Si	Deshabilitar	Seleccionar
Durango	Si	Deshabilitar	Seleccionar

<< < 1 2 3 4 > >>
Ir a página: Registros por página:
Registros: 1 a 10 de 34

Figura C.5: Ejemplo Catálogo - Baja

domingo 13 abr 2014 - Rocio Rodriguez Salir

Inicio > Usu > Alta usuario administrador

Alta usuario administrador

Datos del usuario

Nombre(s)

Apellido paterno

Apellido materno * Correo inválido

Correo electrónico * Este campo es obligatorio

Rol

Autenticación en el sistema

Login

Contraseña Medio * Los campos no coinciden

Repetir Contraseña

Figura C.6: Ejemplo de validaciones en Catálogos

Apéndice D

Exportación de Reportes

	A	B	C	D	E	F	G	H
1	Monitoreo corte semanal y mensual - Estado(s): Guanajuato; Cierre (Semanal) : Semana 20 (del 13/05/2013 al 19/05/2013)							
2	Guanajuato							
3	ABASOLO							
4	ABASOLO							
5	4 estrellas							
6	HOTEL BALNEARIO LA CALDERA							
7	Acámbaro							
8	ACAMBARO							
9	1 estrella							
10	HOTEL DON RAUL							
11	2 estrellas							
12	HOTEL SAN ANTONIO							
13	HOTEL SANTELMO							
14	3 estrellas							
15	HOTEL ACAMBARO PLAZA							
16	HOTEL TERRAZA JARDIN							
17	MOTEL BERMUDAS							
18	4 estrellas							
19	MESON DEL PUENTE							
20	REAL MISION DE ACAMBARO, S.A. DE C.V.							
21	Celaya							
22	CELAYA							
23	1 estrella							
24	HOTEL ACAPULCO TROPICAL I							
25	HOTEL ACAPULCO TROPICAL II							
26	HOTEL R.J							

Figura D.1: Monitoreo de Organismos de Alojamiento, Exportación de datos a hojas de cálculo

		A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	
		Reporte del monitoreo - Centro(s) Turístico(s) : Cozumel, Isla Mujeres; Consulta (Semanal) : Semana 14 (del 01/04/2013 al 07/04/2013)																					
		Cuartos		Cuartos ocupados		Llegada turistas		Turistas noche		Porcentaje de ocupación		Estadía		Densidad									
		Registrados	Disponibles	Total	Residentes	No residentes	Total	Residentes	No residentes	Total	Residentes	No residentes	Total	Residentes	No residentes	Total	Residentes	No residentes					
5	Cozumel																						
6	1 estrella	213	1,463	219	145	74	437	273	164	437	273	164	14,97	9,91	5,06	1	1	1	1	2	1,88	2,22	
7	2 estrellas	186	1,183	142	142	0	284	284	0	284	284	0	12	12	0	1	1	1	0	2	2	0	
8	3 estrellas	526	3,346	803	519	284	819	535	284	2,446	1,620	826	24	15,51	8,49	2,99	3,03	3,03	2,91	3,05	3,12	2,91	
9	4 estrellas	1,225	8,575	5,930	3,562	2,368	3,708	2,408	1,300	13,611	7,857	5,754	69,16	41,54	27,62	3,67	3,26	3,26	4,43	2,3	2,21	2,43	
10	5 estrellas	2,405	16,835	11,652	2,731	8,921	5,169	1,168	4,001	28,992	7,197	21,795	69,21	16,22	52,99	5,61	6,16	6,16	5,45	2,49	2,64	2,44	
11	Total categorías	4,555	31,402	18,746	7,099	11,647	10,417	4,668	5,749	45,770	17,231	28,539	59,7	22,61	37,09	4,39	3,69	3,69	4,96	2,44	2,43	2,45	
12																							
13	Isla Mujeres																						
14	1 estrella	154	1,078	622	622	0	1,158	1,158	0	1,244	1,244	0	57,7	57,7	0	1,07	1,07	1,07	0	2	2	0	
15	2 estrellas	250	1,617	726	312	414	847	583	264	1,452	623	829	44,89	19,29	25,6	1,71	1,07	1,07	3,14	2	2	2	
16	3 estrellas	170	1,162	638	267	371	588	267	321	1,345	549	796	54,91	22,98	31,93	2,29	2,06	2,06	2,48	2,11	2,06	2,15	
17	4 estrellas	280	1,820	1,416	386	1,030	775	354	421	3,034	909	2,125	77,8	21,21	56,59	3,91	2,57	2,57	5,05	2,14	2,35	2,06	
18	5 estrellas	392	2,744	1,831	627	1,204	3,889	1,485	2,404	4,928	1,770	3,158	66,73	22,85	43,88	1,27	1,19	1,19	1,31	2,69	2,82	2,62	
19	Total categorías	1,246	8,421	5,233	2,214	3,019	7,257	3,847	3,410	12,003	5,095	6,908	62,14	26,29	35,85	1,65	1,32	1,32	2,03	2,29	2,3	2,29	
20																							
21	Totales																						
22	1 estrella	367	2,541	841	767	74	1,595	1,431	164	1,661	1,517	164	33,09	30,18	2,91	1,05	1,06	1	2	1,88	2,22		
23	2 estrellas	436	2,800	868	454	414	1,131	867	264	1,736	907	829	31	16,21	14,79	1,53	1,05	1,05	3,14	2	2	2	
24	3 estrellas	696	4,508	1,441	786	655	1,407	802	605	3,791	2,169	1,622	31,97	17,44	14,53	2,69	2,7	2,7	2,68	2,63	2,76	2,48	
25	4 estrellas	1,505	10,395	7,346	3,948	3,398	4,483	2,762	1,721	16,645	8,766	7,879	70,67	37,98	32,69	3,71	3,17	3,17	4,58	2,27	2,22	2,32	
26	5 estrellas	2,797	19,579	13,483	3,358	10,125	9,058	2,653	6,405	33,920	8,967	24,953	68,86	17,15	51,71	3,74	3,38	3,38	3,9	2,52	2,67	2,46	
27	Total	5,801	39,823	23,979	9,313	14,666	17,674	8,515	9,159	57,773	22,326	35,447	60,22	23,39	36,83	3,27	2,62	2,62	3,87	2,41	2,4	2,42	

Figura D.2: Monitoreo de Organismos de Alojamiento, Exportación de datos a hojas de cálculo

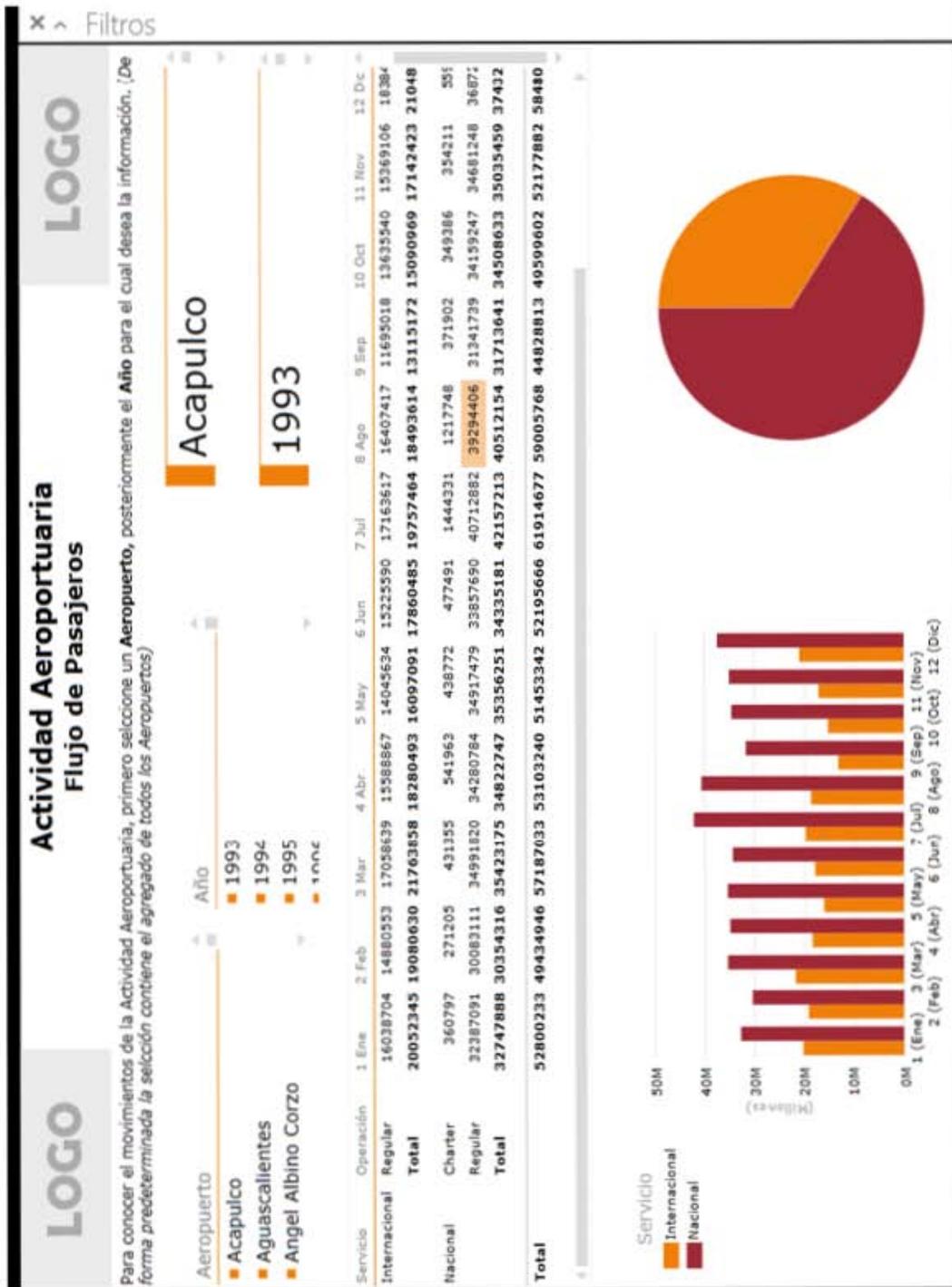


Figura D.3: Ejemplo de Reportes - Actividad Aeroportuaria

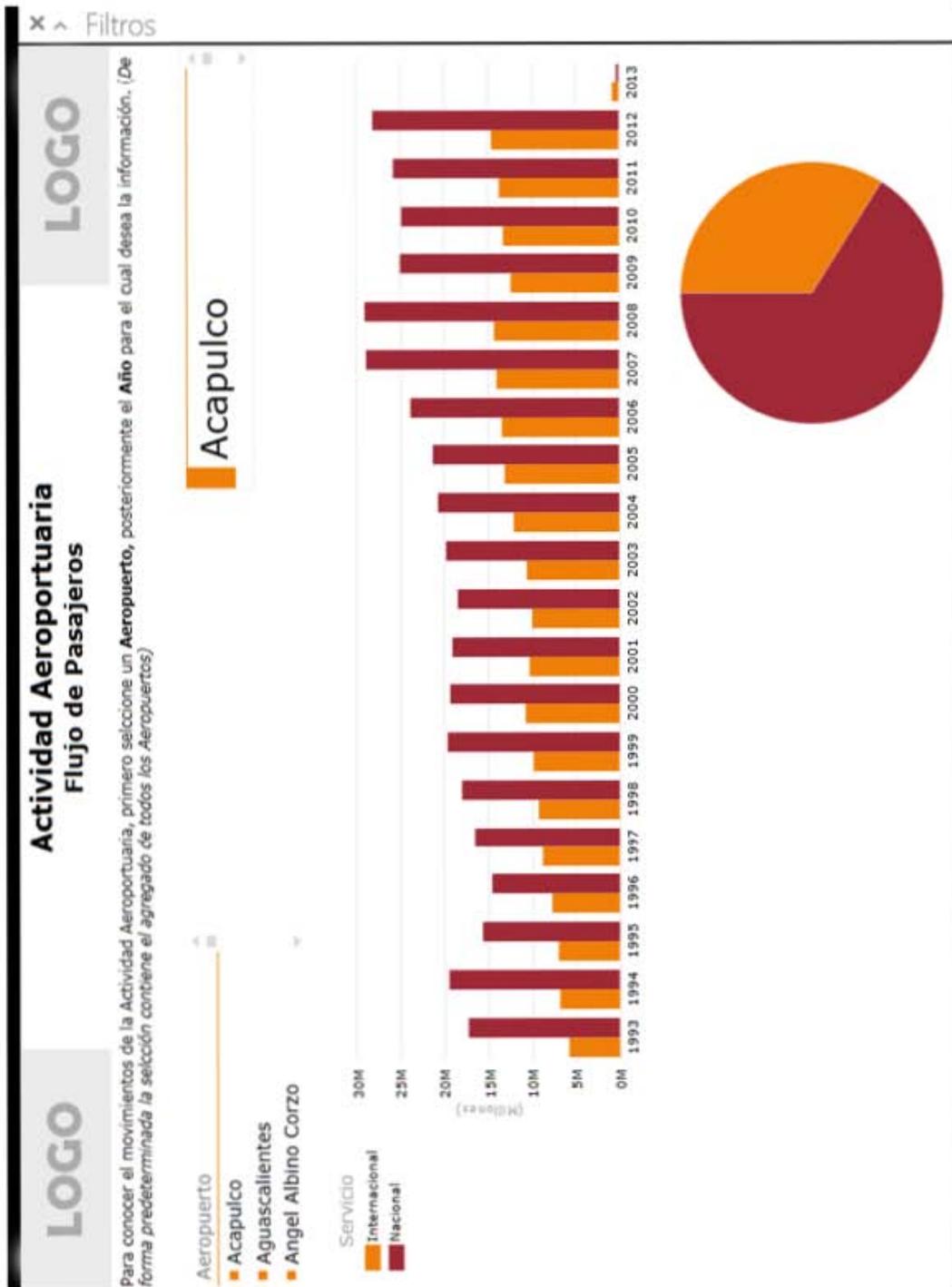


Figura D.4: Ejemplo de Reportes - Actividad Aeroportuaria

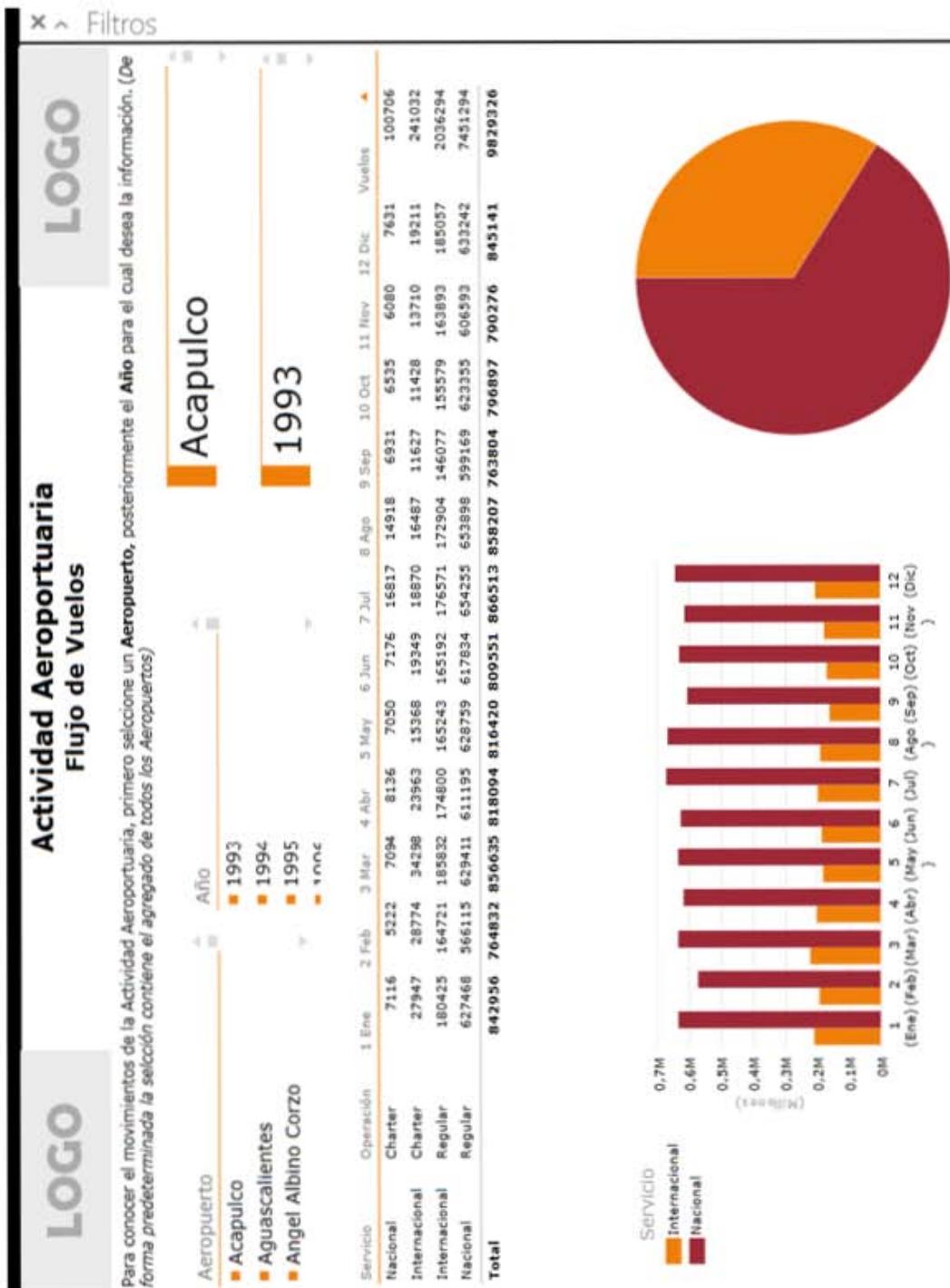


Figura D.5: Ejemplo de Reportes - Actividad Aeroportuaria

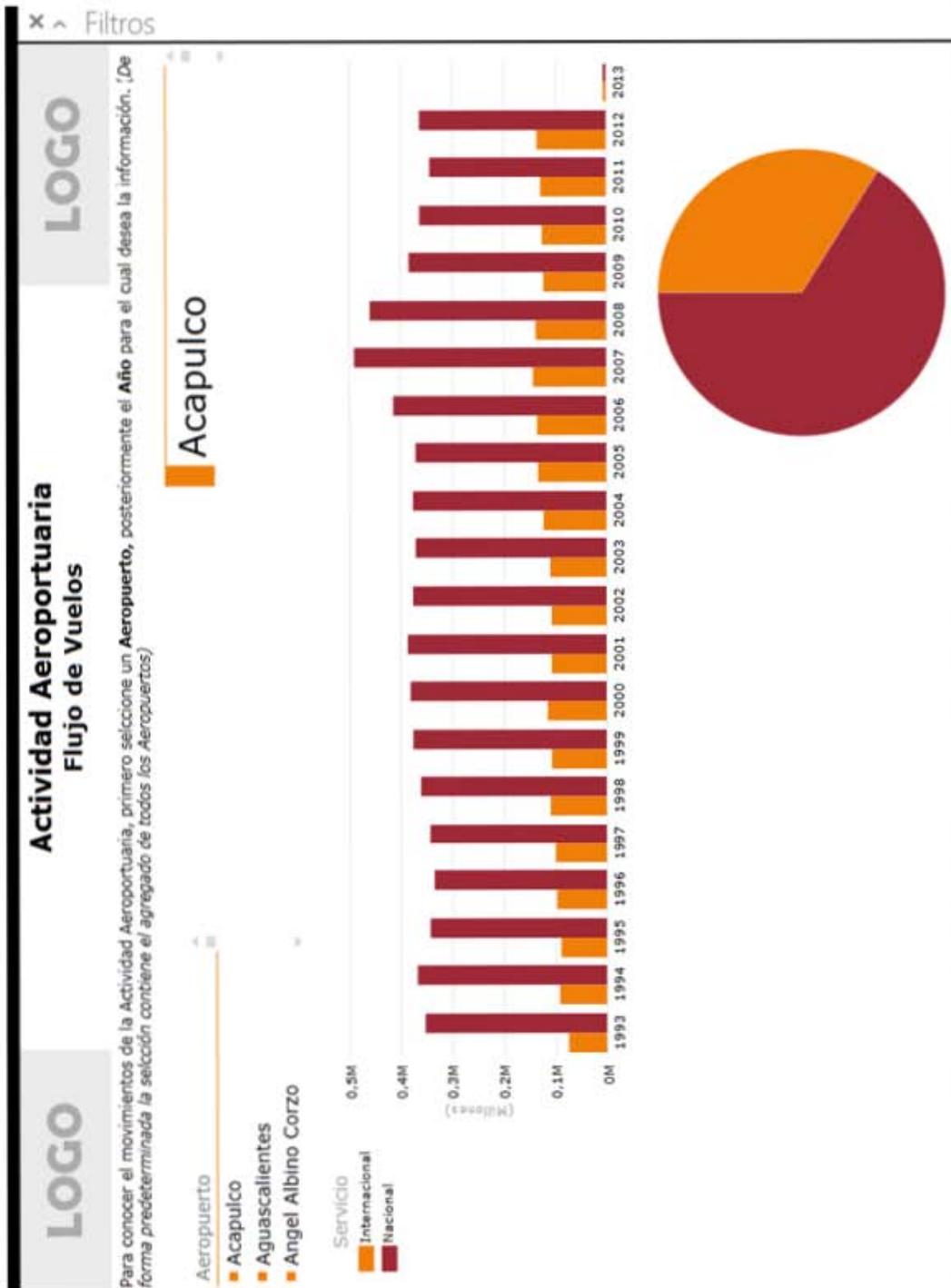


Figura D.6: Ejemplo de Reportes - Actividad Aeroportuaria



Figura D.7: Ejemplo de Reportes - Conectividad Aérea, Servicio Doméstico

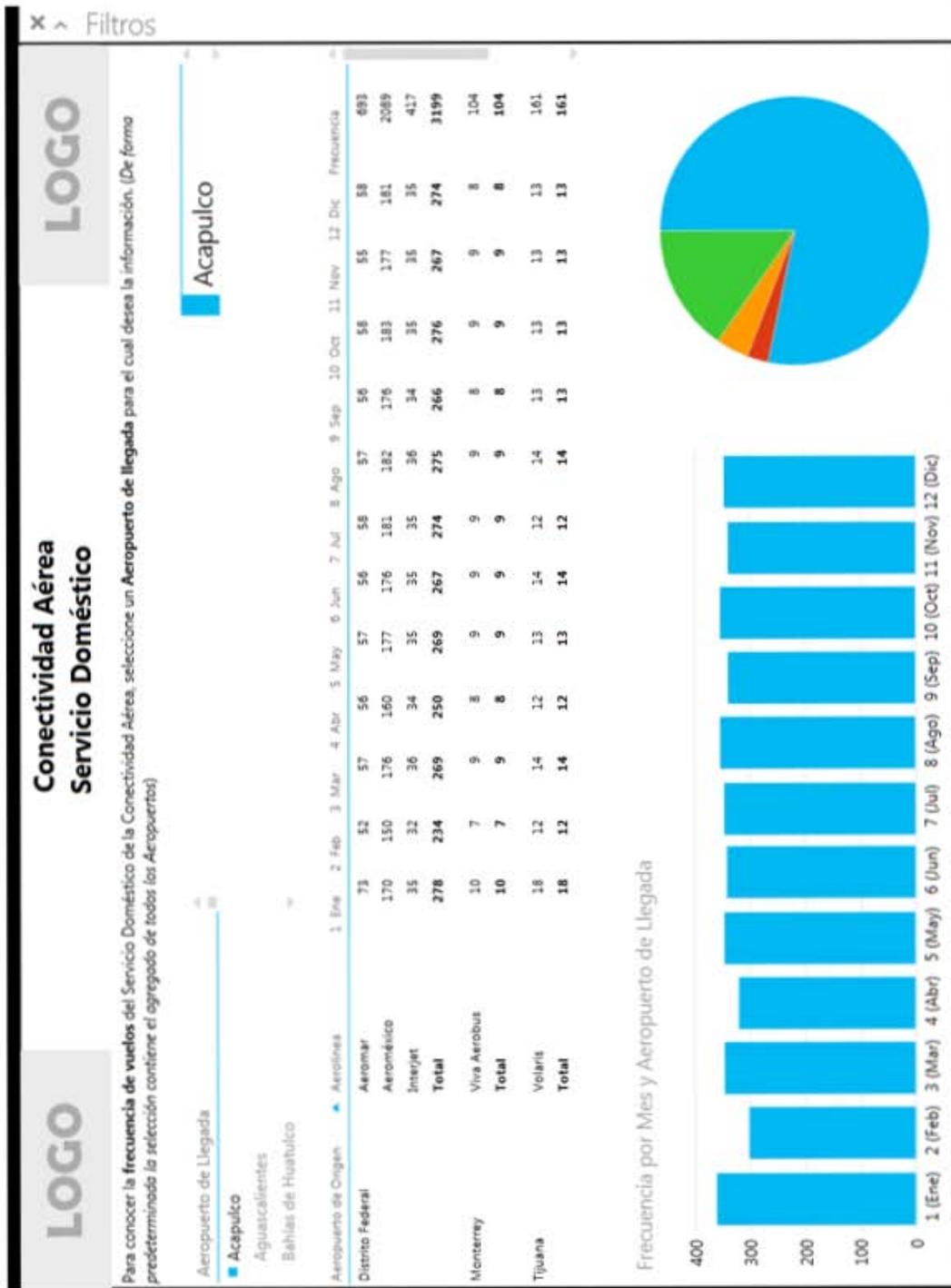


Figura D.8: Ejemplo de Reportes - Conectividad Aérea, Servicio Doméstico

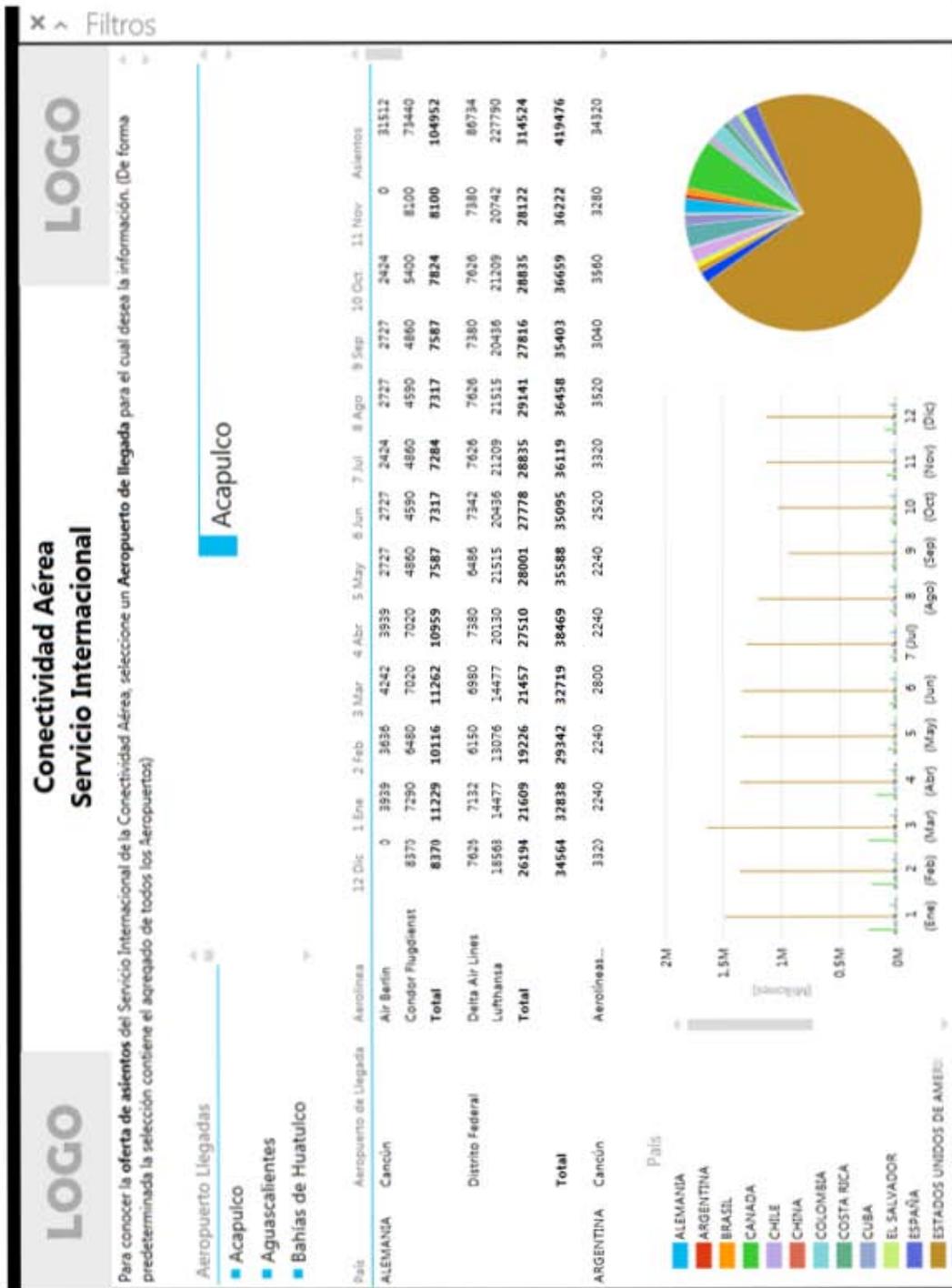


Figura D.9: Ejemplo de Reportes - Conectividad Aérea, Servicio Internacional

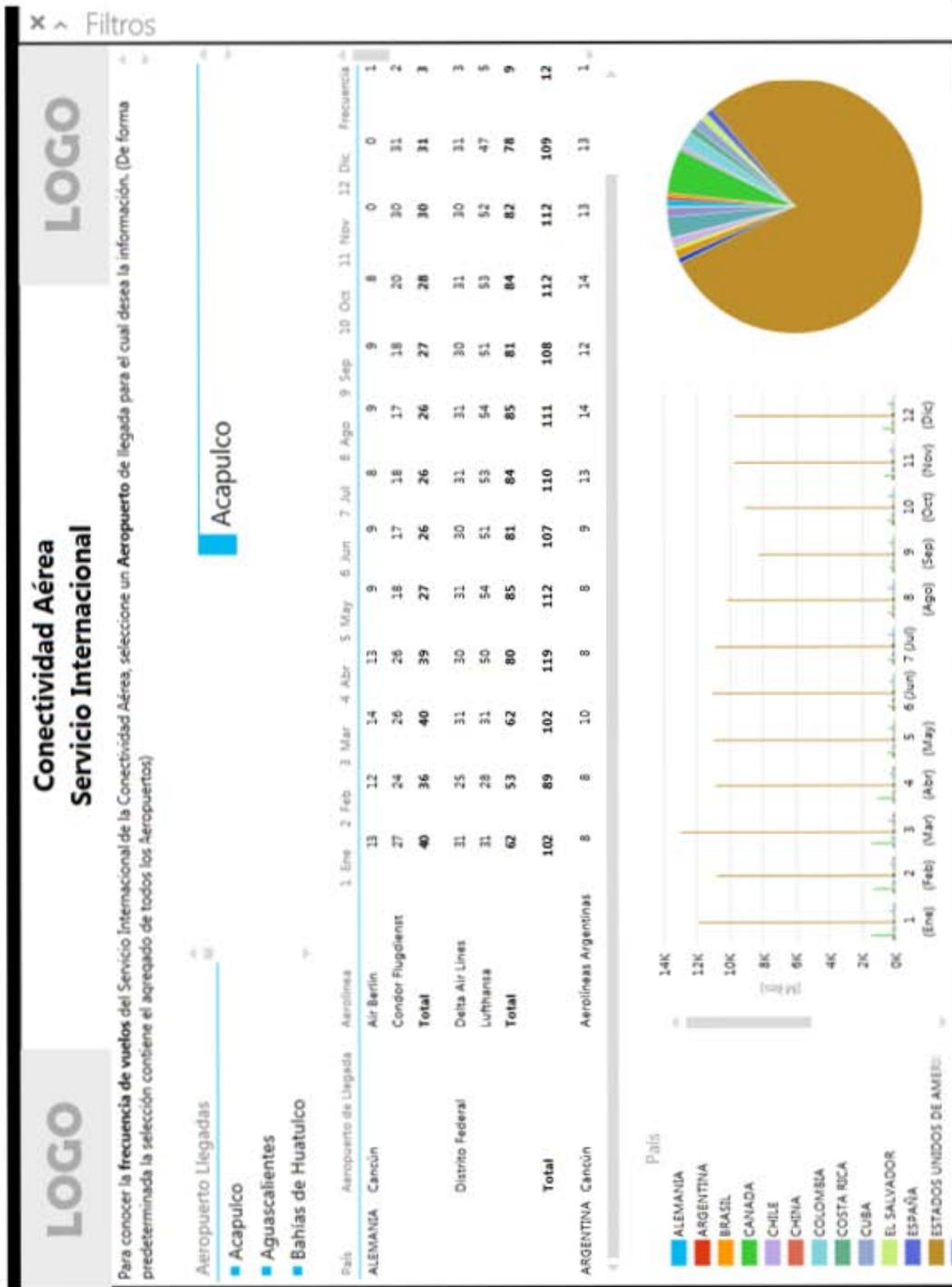


Figura D.10: Ejemplo de Reportes - Conectividad Aérea, Servicio Internacional

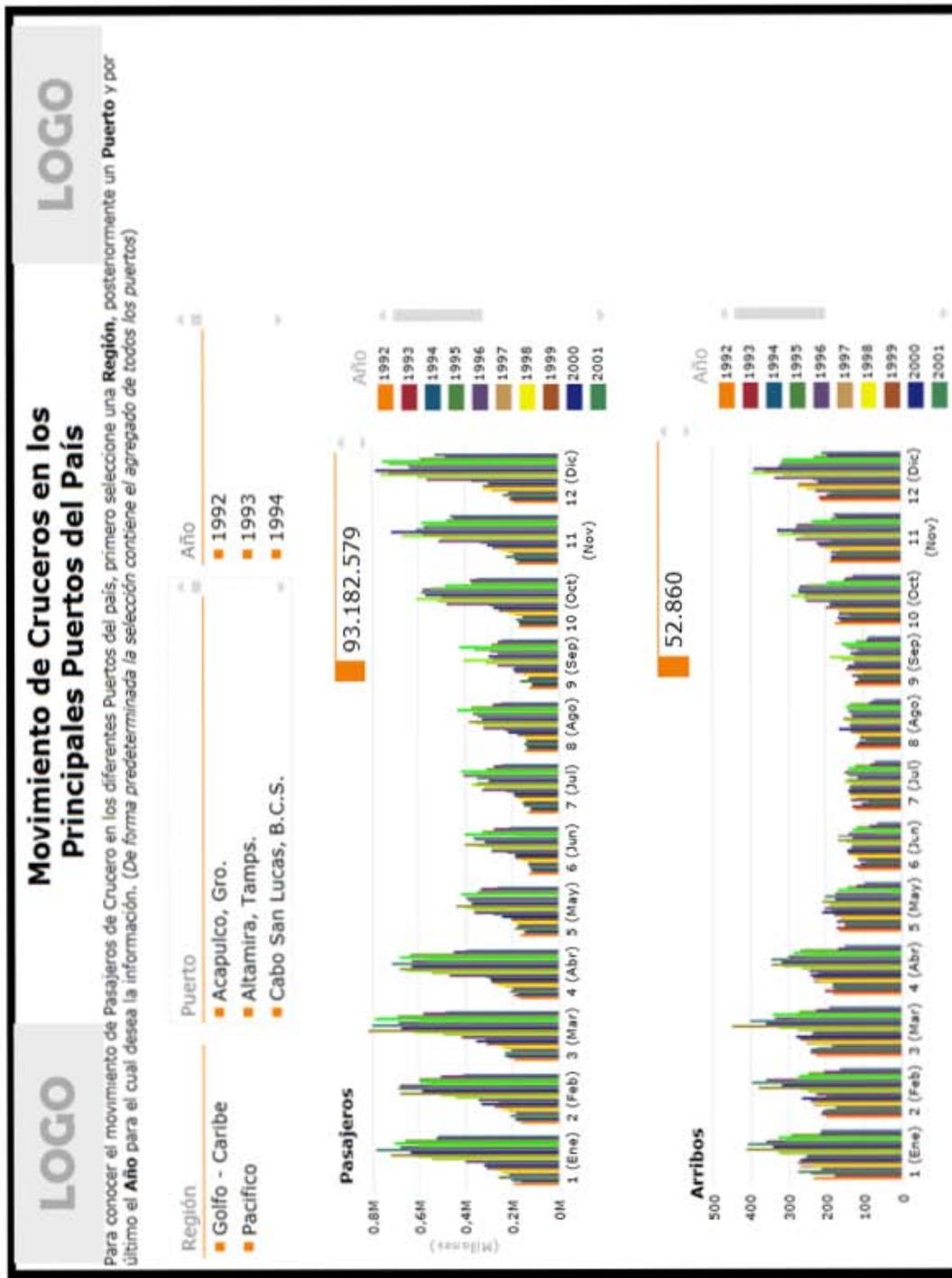


Figura D.11: Ejemplo de Reportes - Movimiento de Cruceros

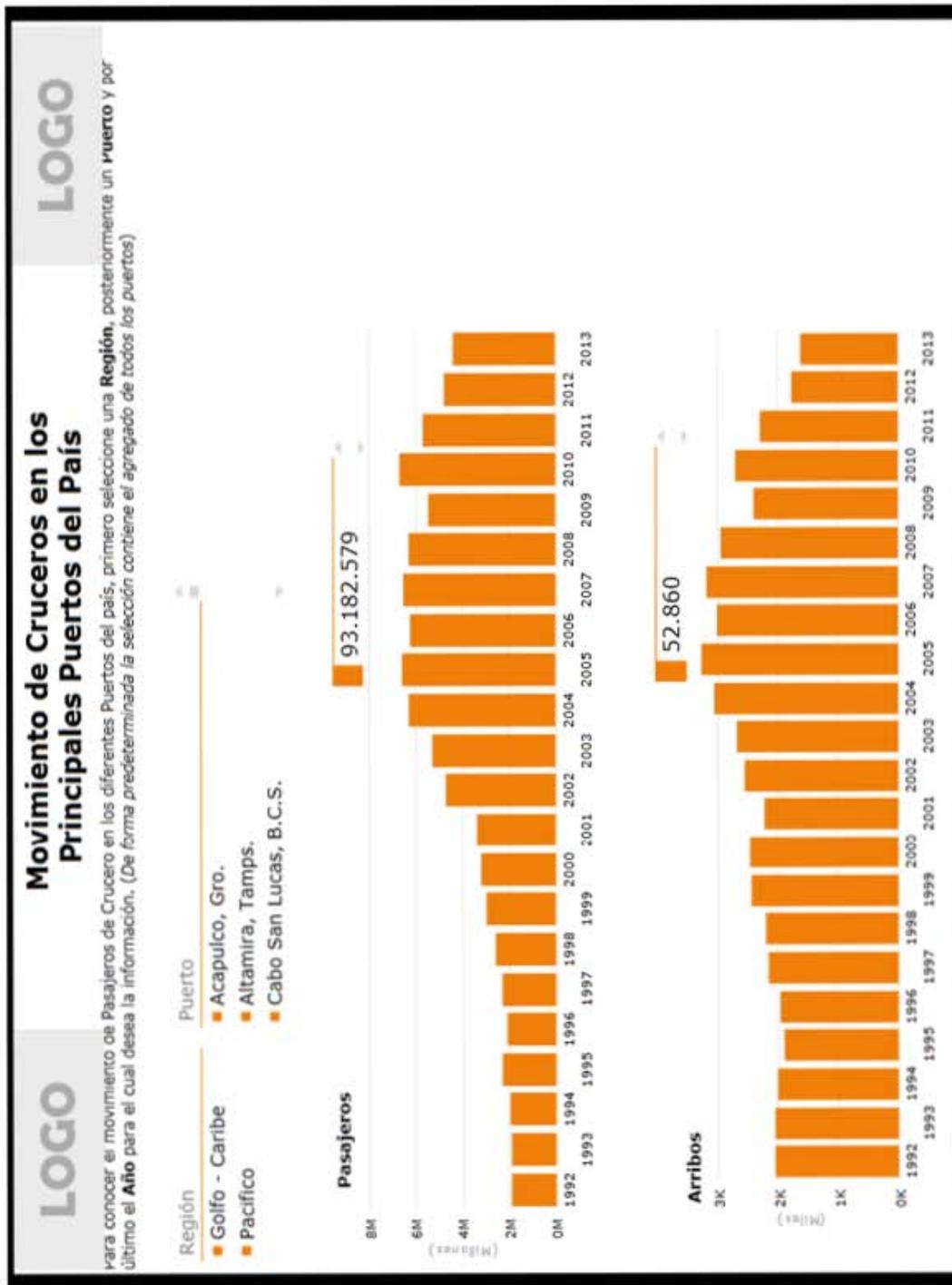


Figura D.12: Ejemplo de Reportes - Movimiento de Cruceros

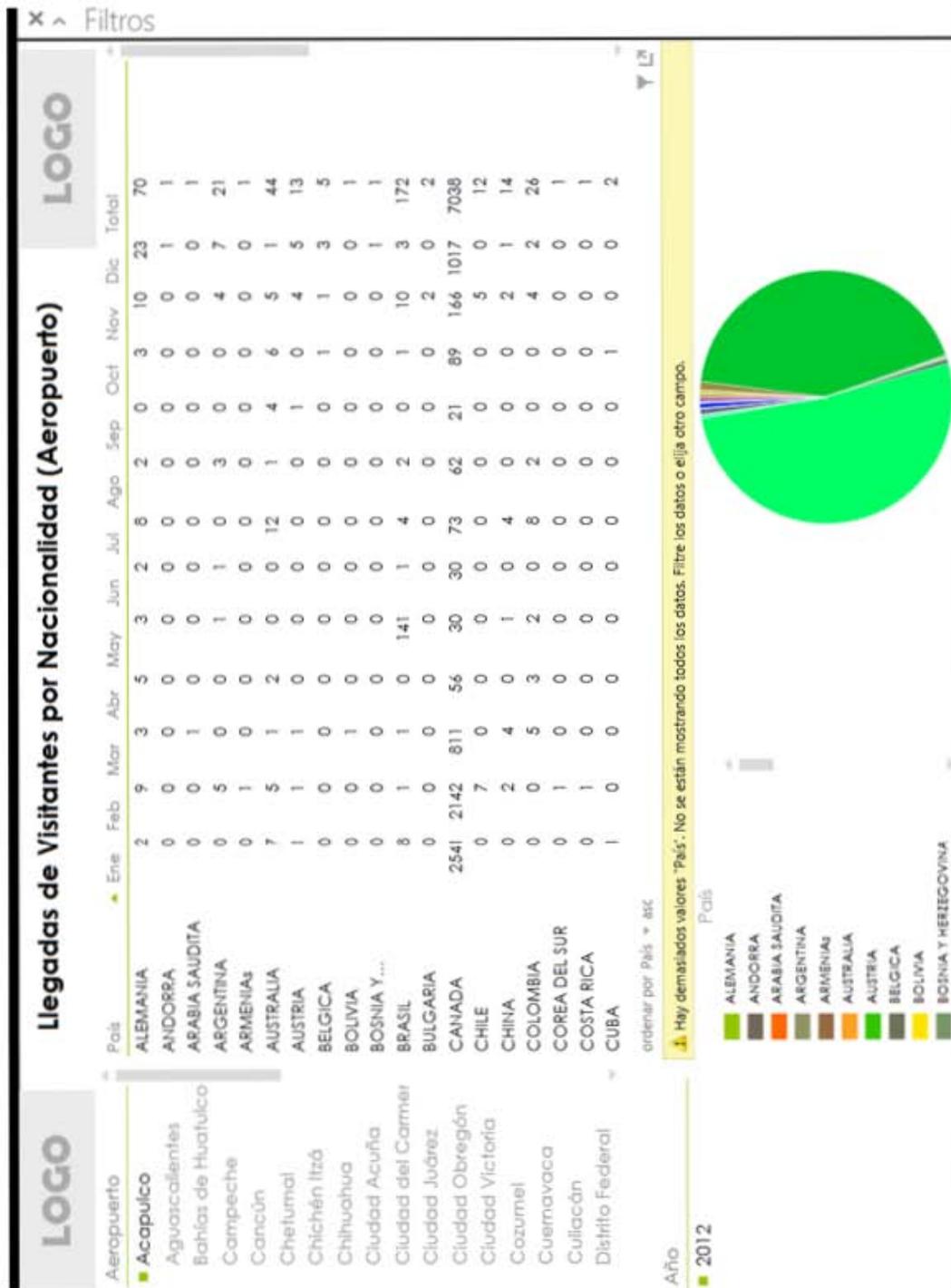


Figura D.13: Ejemplo de Reportes - Llegada de Visitantes por Nacionalidad

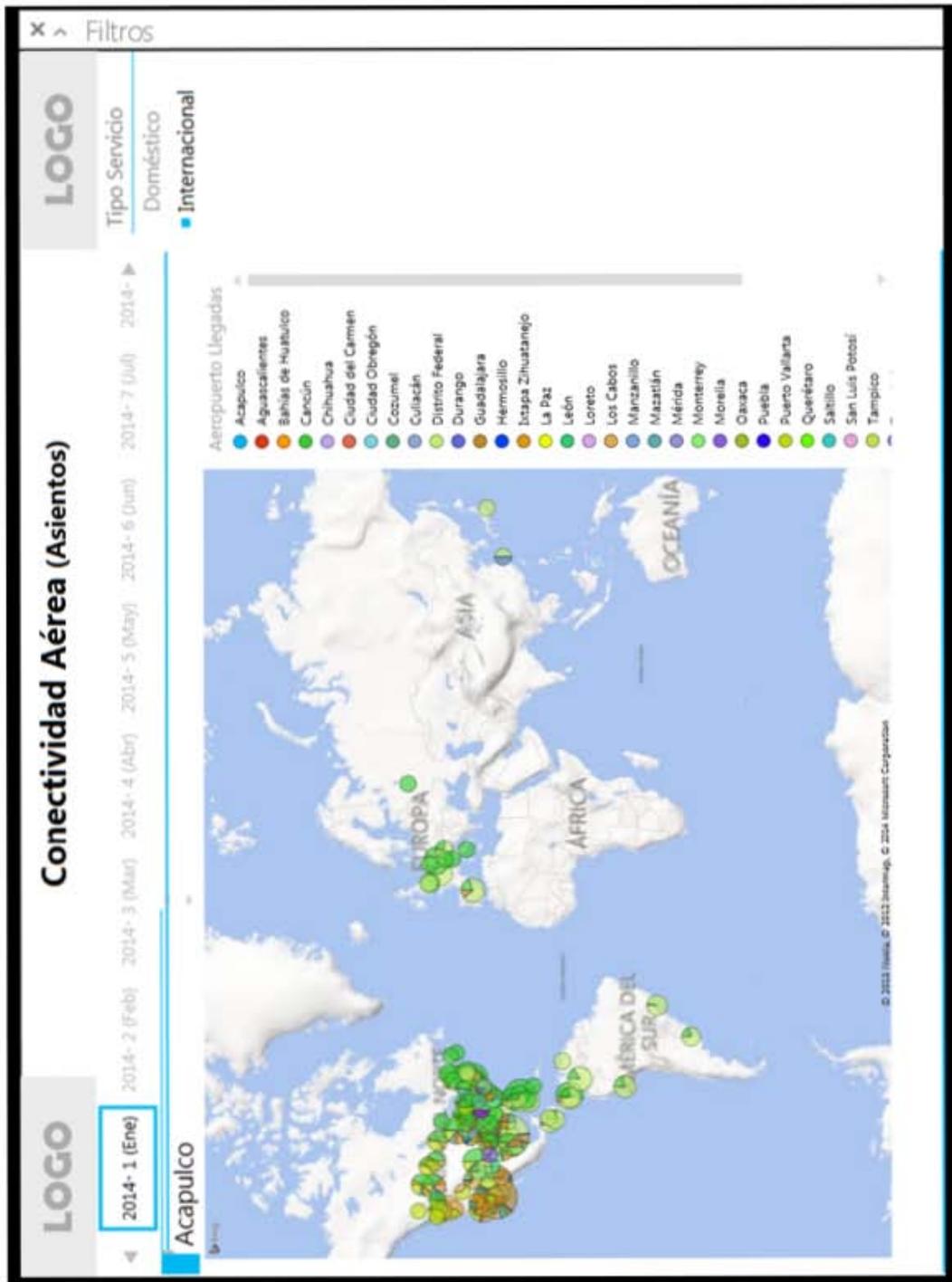


Figura D.14: Ejemplo de Reportes - Conectividad Aérea (asientos)

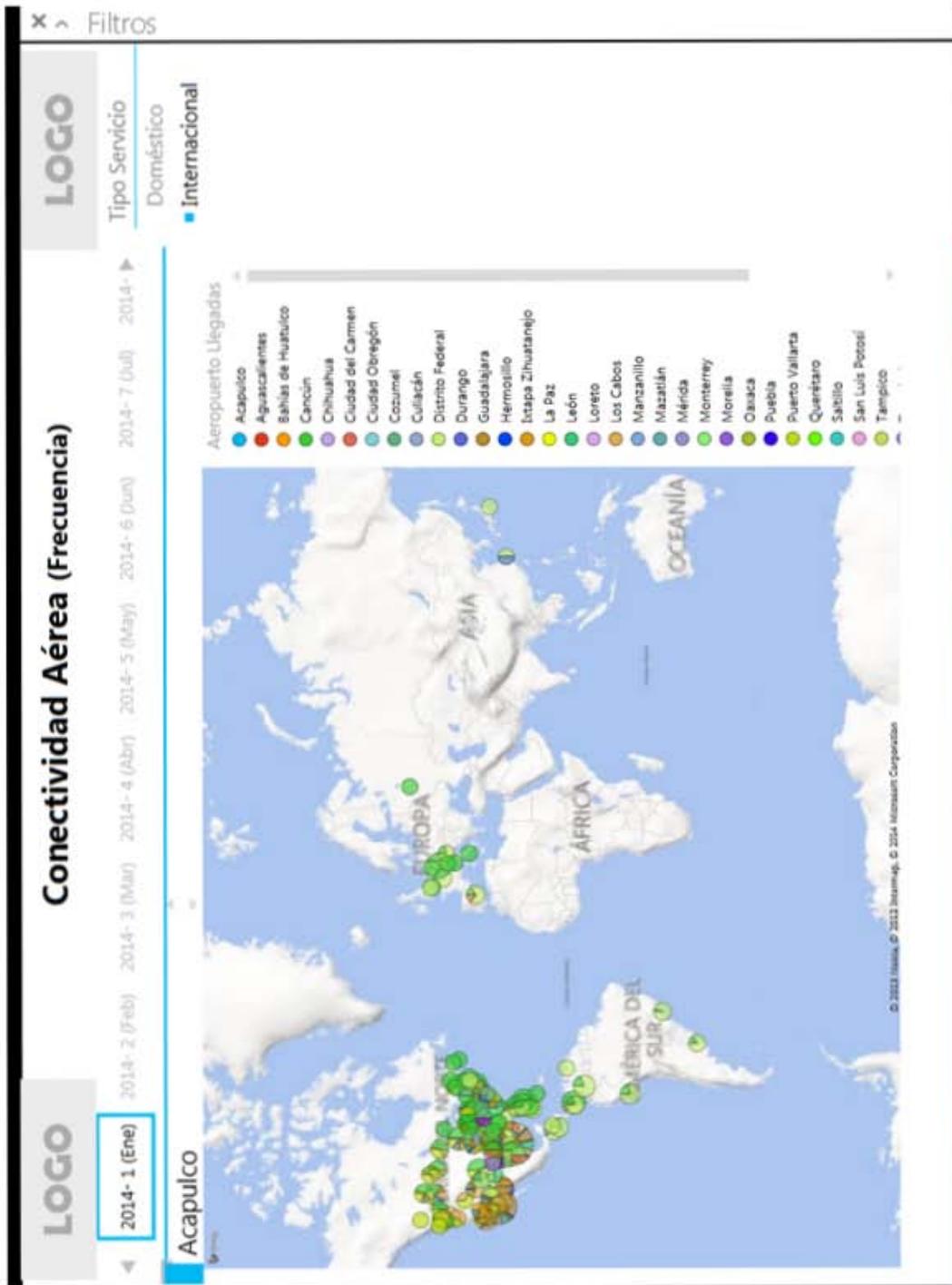


Figura D.15: Ejemplo de Reportes - Conectividad Aérea (frecuencia)

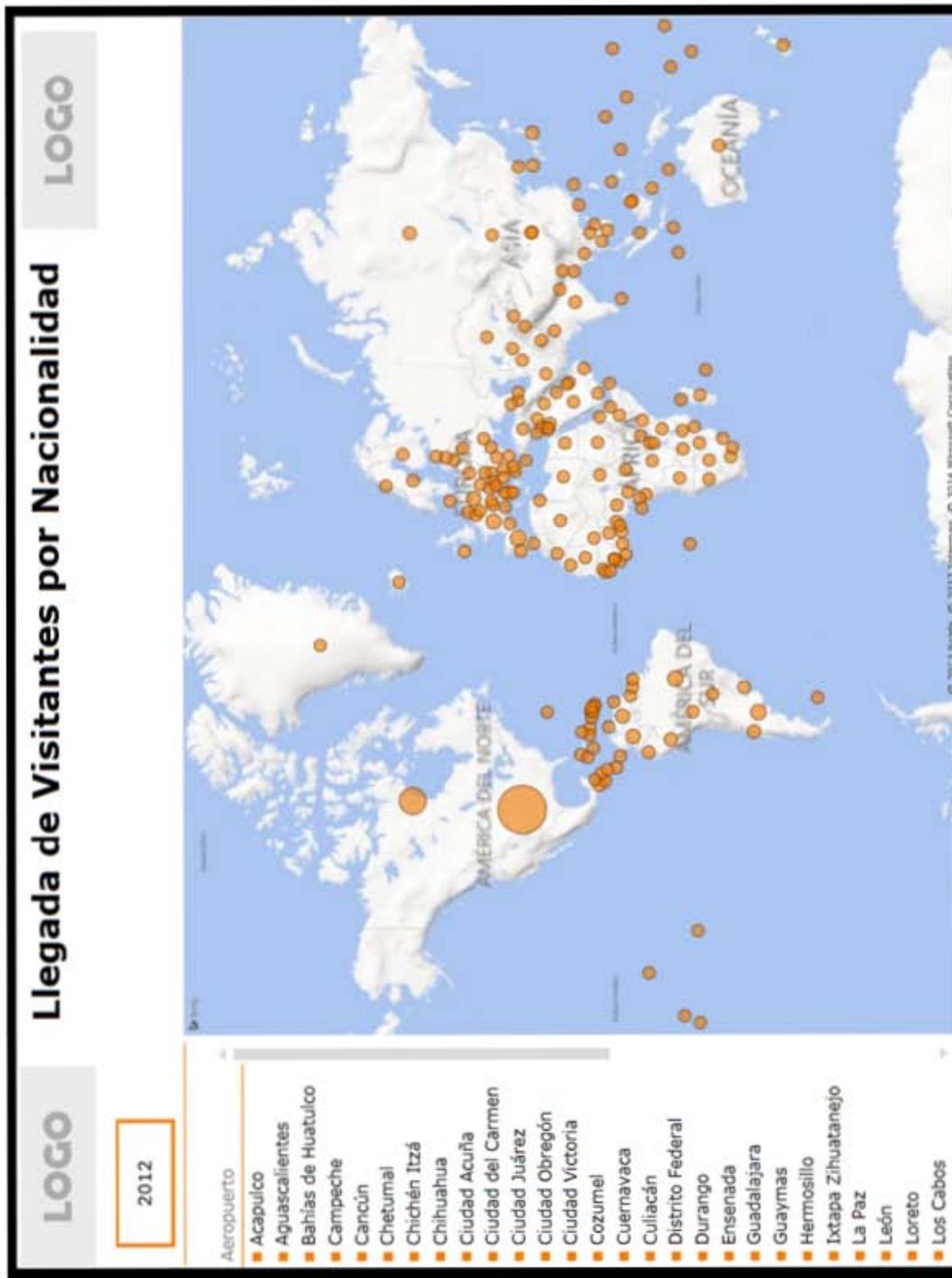


Figura D.16: Ejemplo de Reportes - Llegada de Visitantes por Nacionalidad

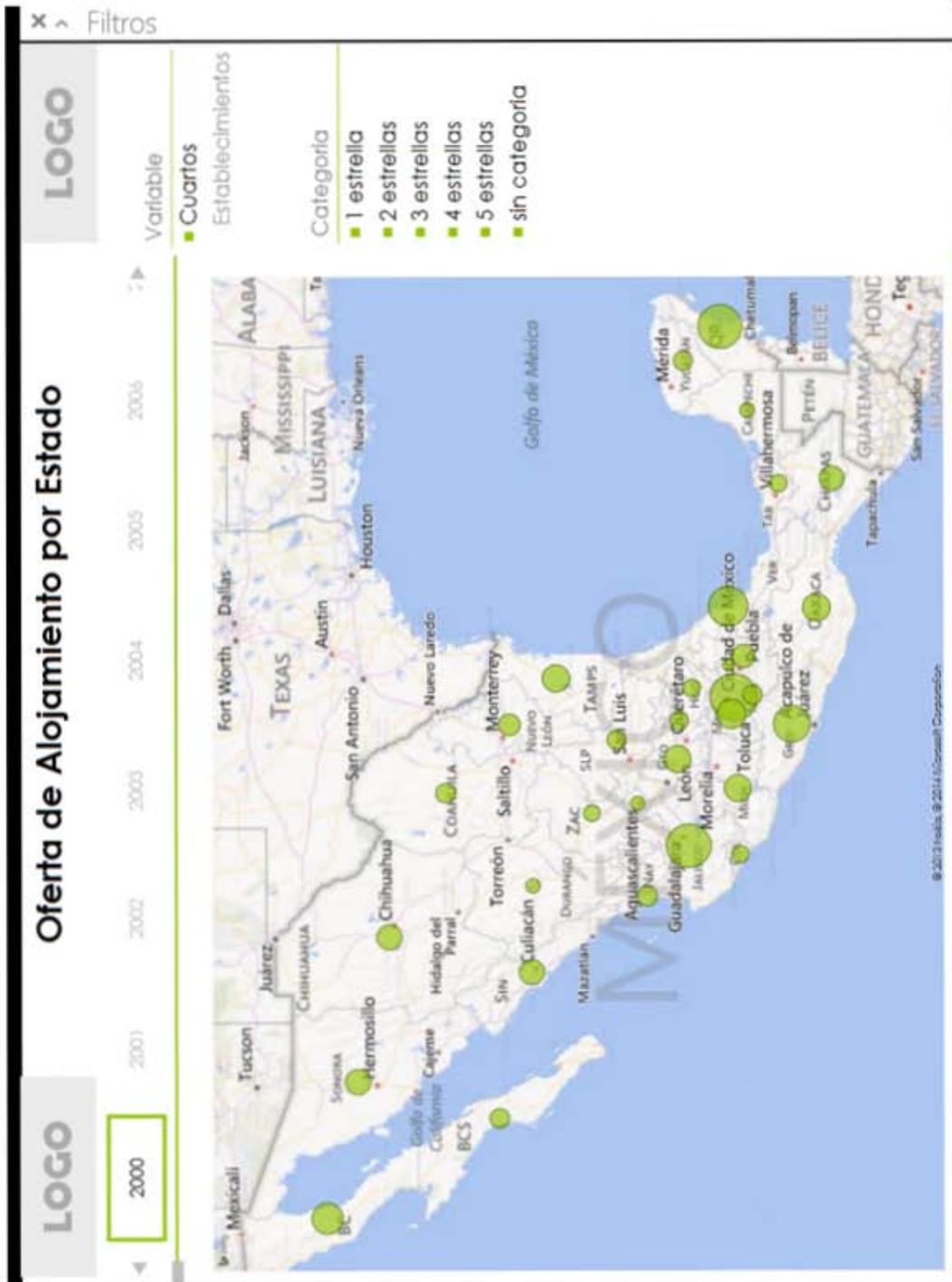


Figura D.17: Ejemplo de Reportes - Oferta de Alojamiento por Estado



Figura D.18: Ejemplo de Reportes - Viajeros Internacionales a México

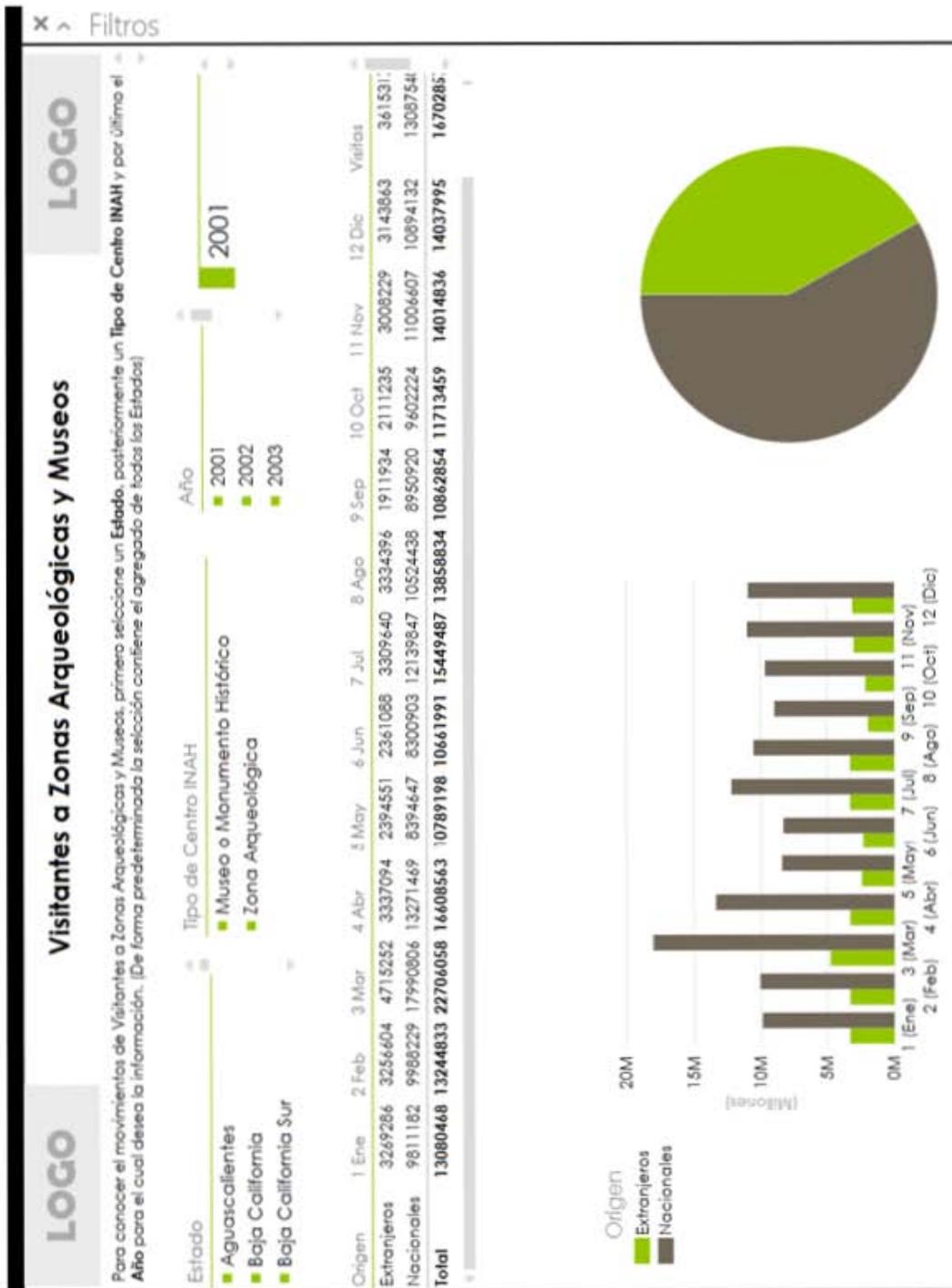


Figura D.19: Ejemplo de Reportes - Visitantes a Zonas Arqueológicas y Museos

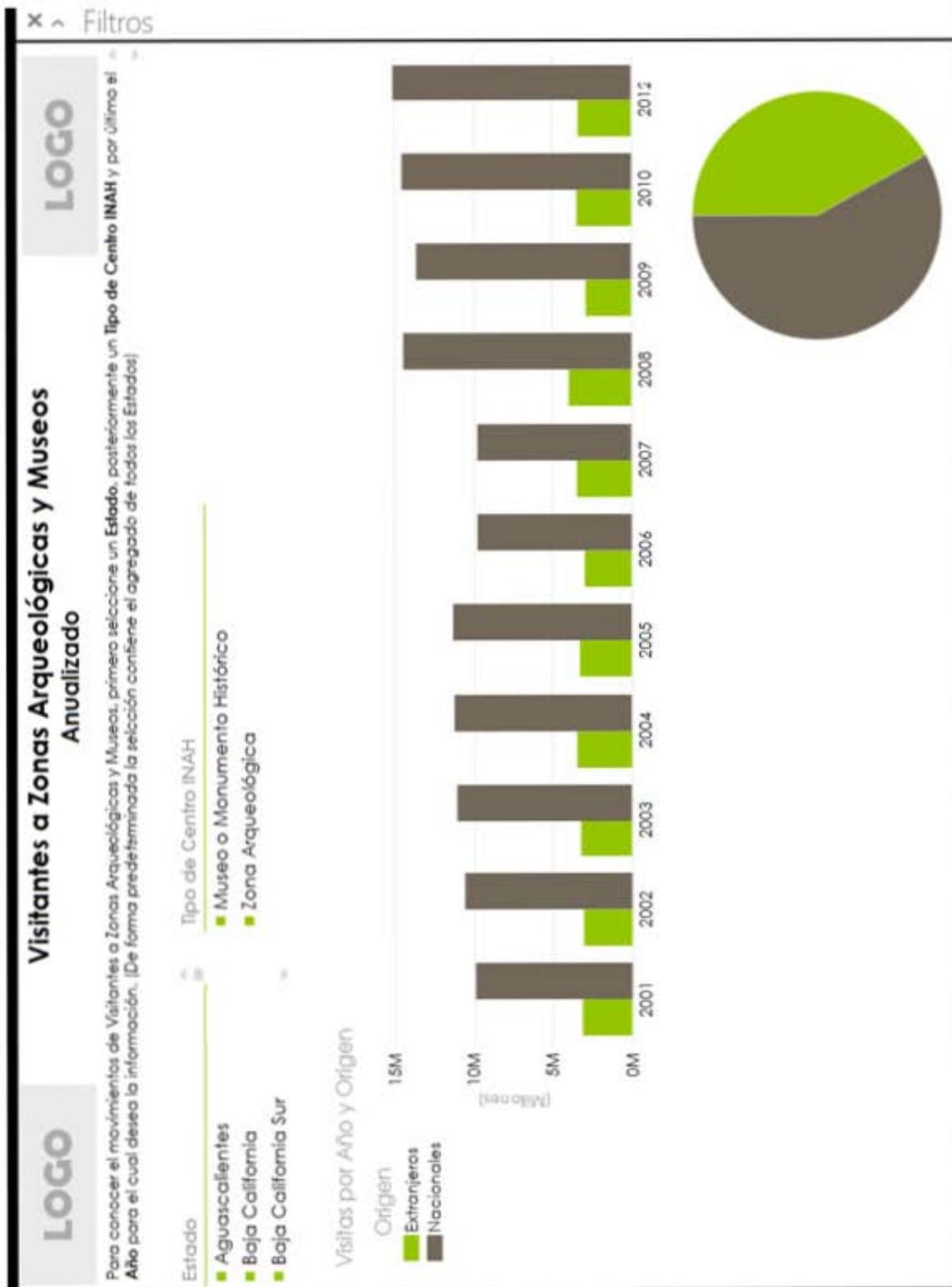


Figura D.20: Ejemplo de Reportes - Visitantes a Zonas Arqueológicas y Museos (anualizado)

Referencias

- [1] Jiawei Han, Micheline Kamber, *Data Mining Concepts and Techniques*. Third Edition, Elsevier, USA 2012
- [2] García Molina, Hector, *Database Systems: The Complete Book* Prentice Hall, New Jersey 2002
- [3] John E. Hopcroft, Rajeev Motwani, Jeffrey Ullman, *Introduction to Automata Theory, Languages and Computation*. Second Edition, Addison-Wesley Publishing Company, Pearson 2001
- [4] Tamraparni Dasu & Theodore Johnson, *Exploratory Data Mining and Data Cleaning*. John Wiley & Sons, New Jersey 2003
- [5] Ling Liu & M. Tamer Ozsü, *Encyclopedia of Database System*. Springer, USA 2009