Universidad Nacional Autónoma de México

Facultad de Ciencias

Desarrollo de software de propósito específico para el programa Coleccionando Sonrisas de la Fundación para la Protección de la Niñez I.A.P.

REPORTE DE TRABAJO PROFESIONAL

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

PRESENTA:
ODÍN MIGUEL ESCORZA SORIA

DIRECTOR DE TESIS:
M. EN A. KARLA RAMÍREZ PULIDO

2014

Agradecimientos

Como no hay manera, para mí, de escribir esta sección sin hacer omisiones ni cometer injusticias emperezaré por ofrecer una gran disculpa a todas las personas que no mencione en el texto siguiente aclarando que la ausencia de su nombre responde únicamente a mi memoria de teflón. A todos ellos muchísimas gracias por hacer posible, cada quien a su manera, que completara mi licenciatura por un lado y me titulara después de tanto tiempo por el otro.

Diría mi madre, parafraseando el corrido *Benito Canales*: ahora que estoy disculpado, quiero escribir otro rato...

Este trabajo no habría sido posible sin todo el apoyo y motivación que recibí de mi tutora Mtra. Karla Ramírez quien desde que comenzó a dirigir mis diversos proyectos de titulación (sí, este fue el que cuajó pero hubo por lo menos otros dos) jamás dejó de confiar en que llegaría este momento y estuvo ahí, siempre dispuesta a guiarme y ayudarme.

No hay muestra de gratitud suficiente para reconocer el valor de las diversas observaciones y correcciones que mis sinodales Mtra. Karla, Dra. Amparo, Dra. María de Luz, Dr. José y Lic. Sergio hicieron sobre mi trabajo permitiéndome no sólo mejorar el texto sino mi formación como profesional y como persona. Valor que se multiplica por todo el conocimiento que me dejaron adquirir a través de sus cursos que dejaron huellas impresas en mí para toda la vida.

Agradezco especialmente mi jefe Mauricio Aguilar por todo su apoyo, por toda la confianza que ha depositado en mí y en mi trabajo, por todas su enseñanzas y por tantas y tantas horas de reflexión, buena música e invaluables consejos. Porque ha estado dispuesto a todo con tal de verme crecer a pesar de todos los infortunios que le han sucedido a lo largo del tiempo que nos conocemos.

A mi mamá muchas gracias por enseñarme esta pasión por la docencia que fue uno de los motivos más fuertes que me llevaron a concluir este ciclo en mi vida académica.

Papá, gracias por enseñarme que la perseverancia y la humildad son virtudes que abren puertas aún en los contextos más adversos.

Sinuhé, en los últimos dos años tus experiencias me han permitido aprender, primero, que la rectitud, la bondad y las ganas de ayudar a otros pueden prevalecer aún cuando nos encontramos en las circunstancias más complicadas. Y segundo, que el camino de la rectitud no nos lleva al mismo lugar cuando está pavimentado de resentimiento, miedo e ignorancia. Estas dos lecciones han quedado impresas en mi alma para siempre gracias a ti, hermano.

A Beatriz, porque has estado conmigo en una etapa de grandes cambios y fuertes dificultades. Porque con tu cariño, comprensión y apoyo incondicional me has dado el regalo de ayudarme a descubrir riquezas que estaban escondidas en mí desde hace muchos años.

A mis amigos Alma, Elisa, Karla, Alberto, Ernesto, Gustavo, Israel, Marco y Ricardo que a su modo contribuyeron haciendo presión para que por fin me animara a titularme, muchas gracias.

A mi maestro Roberto Olivera Mondragón, que ha sido inspiración y ejemplo de lo que debe ser un docente durante toda mi experiencia como ayudante en la Facultad.

Por último, quiero agradecer a todas las personas a quienes he lastimado a lo largo de mi vida pues su perdón y comprensión me han permitido continuar con mi desarrollo. A ellos les debo en gran medida y les dedico este trabajo.

Introducción

Según los indicadores de pobreza multidimensional del Instituto Nacional de Estadística y Geografía Informática, INEGI, en 2010 alrededor del 54 % de los niños mexicanos de 0 a 17 años se encontraban en situación de pobreza extrema o moderada[24] y esta tendencia está creciendo, lo cual implica que menos de la mitad de la población infantil en nuestro país tiene actualmente los elementos suficientes para ejercer sus derechos a plenitud y llevar una vida digna.

Algunas Instituciones de Asistencia Privada, IAP, son un medio para contribuir al desarrollo social y en particular al mejoramiento de esta condición de pobreza en los niños de nuestro país. Sin embargo, a pesar de que muchas de ellas cuentan con recursos económicos suficientes para promover y desarrollar programas de apoyo, difícilmente logran vincularse con instituciones o empresas que les provean de herramientas tecnológicas[1] adecuadas para llevarlos a cabo de manera eficaz. Como consecuencia de esta falta de vinculación, las instituciones se ven forzadas a gestionar sus iniciativas de forma manual o, en los mejores casos, con documentos digitalizados, situación que merma los resultados de las mismas debido, entre otras cosas, a:

- La falta de solidez en el almacenamiento de datos.
- Lentitud en procesos de comunicación.
- Complejidad en la administración de recursos.
- Falta de control sobre flujos de información.

Una solución potencial a la merma de resultados en las iniciativas de las IAP es el desarrollo de software de propósito específico[2]. En el presente texto se expondrá la experiencia profesional y la aplicación de los conocimientos adquiridos como

[1] software, computadoras, redes y demás.

[2] También llamado *software Ad-Hoc*, es decir, software fabricado según los requerimientos particulares del cliente o usuario final.

egresado de Ciencias de la Computación en el desarrollo de un conjunto de herramientas de software de propósito específico desarrolladas para mejorar el programa *Coleccionando Sonrisas* de la *Fundación para la Protección de la Niñez* (FPN) una *Institución de Asistencia Privada*.

El primer capítulo presenta información introductoria sobre la *Fundación* y el programa *Coleccionando Sonrisas*. También describe el escenario, en términos de sistemas computacionales, en el que se encontraba el programa al inicio del proyecto y los objetivos generales del mismo permitiendo al lector tener una visión más amplia acerca del contexto en el que se realizó este trabajo.

En el capítulo segundo se describen con detalle los objetivos concretos y requerimientos funcionales del sistema desarrollado para dar a conocer las necesidades que se pretendía satisfacer al inicio del proyecto.

Se presenta en el tercer capítulo una descripción detallada del modelo de las entidades relacionadas con la aplicación, además de la implementación de la base de datos, esto es, la representación de dichas entidades como tablas relacionales.

El capítulo cuarto versa sobre el diseño de la aplicación fuera del contexto de la persistencia enfatizando el diseño del algoritmo de distribución mensual de puntos.

El quinto capítulo expone brevemente las herramientas de software que facilitaron y/o condicionaron el proyecto introduciendo al lector en el contexto técnico en el que fue realizado el trabajo.

En el capítulo sexto y último se presentan las conclusiones y las propuestas a futuro para mejorar el sistema desarrollado.

Índice general

Índice de figuras

Índice de Tablas

Capítulo 1

Antecedentes

En este capítulo introductorio se expone el contexto dentro del cual surge el programa *Coleccionando Sonrisas* y el funcionamiento del mismo en términos generales. Las secciones dentro de este capítulo versan sobre la historia, visión y misión de la fundación, sobre el funcionamiento del programa en un bosquejo del análisis de requerimientos y sobre el objetivo de involucrar a un pasante en Ciencias de la Computación en el desarrollo del sistema.

1.1 *Fundación para la Protección de la Niñez, IAP*

La *Fundación para la Protección de la Niñez, IAP*, *FPN*, fue creada el 11 de diciembre de 1990 por el Sr. Don David L. Romero Casas con la única finalidad de mejorar las condiciones de vida de niños y jóvenes que se encuentran en un estado de abandono o pobreza extrema, que necesitan del apoyo social para lograr un crecimiento y desarrollo humano estable, [23].

Situación vulnerable

Situación vulnerable es una condición social de riesgo o dificultad que inhabilita, de manera inmediata o en el futuro, a los grupos o individuos afectados en la satisfacción de sus necesidades (en tanto subsistencia y calidad de vida) en contextos socio históricos y culturales determinados, [17].

Pobreza

El Coneval[1], por su parte, define la pobreza de la siguiente manera: "*La pobreza, en su acepción más amplia, está asociada a condiciones de vida que vulneran la dignidad de las personas, limitan sus derechos y libertades fundamentales, impiden la satisfacción de sus necesidades básicas e imposibilitan su plena integración social*"[3].

Donadores y voluntarios

En el contexto de este trabajo un donador es una persona o institución que entrega a la *FPN*, dinero o cualquier otro objeto de valor recibiendo a cambio un recibo de donativo deducible de impuestos. Por otro lado, un voluntario es cualquier persona que desempeña actividades, de manera regular o esporádica, dentro de la *Fundación* sin percibir un salario.

1.2 Programa *Coleccionando Sonrisas*

Usando los recursos que la *FPN* obtiene a través de sus redes de donadores y voluntarios se crean y fortalecen programas diseñados para beneficiar a niños de toda la República Mexicana que se encuentran en situación vulnerable.

Además de las aportaciones económicas, la *Fundación* recibe como donativo cualquier objeto de valor de manera indiscriminada. Desde el punto de vista de la *FPN*, muchos de estos bienes[2] no se pueden entregar directamente a los niños en situación vulnerable y se han acumulado en la bodega de la *Fundación* desde 1990. Al ser donativos, dichos objetos no se pueden vender ni intercambiar para conseguir financiamiento u otros objetos que sí sean beneficios directos considera un beneficio directo cualquier objeto o recurso del cual un niño en situación vulnerable pueda sacar provecho de manera inmediata y sin recurrir a intermediarios. Ejemplos de estos son: alimentos, ropa, calzado, vivienda y demás para los infantes.

Como una solución al problema de acumulación de bienes en la bodega de la *Fundación* surge el programa *Coleccionando Sonrisas* que consiste en un esquema en

[1] Consejo Nacional de Evaluación de la Política de Desarrollo Social

[2] Ejemplos de estos bienes son: juegos de video, electrodomésticos, equipo de laboratorio, etcétera

el que los donadores se registran para hacer una contribución monetaria fija mensual a cambio de una determinada cantidad de puntos. Estos puntos son intercambiables por bienes como teléfonos celulares, electrodomésticos, herramientas y demás que almacena la *FPN* y no puede entregar a los niños en situación vulnerable.

1.2.1 Funcionamiento del programa *Coleccionando Sonrisas*

Un donador se registra en el programa con tarjeta de débito, clave bancaria estandarizada (CLABE) o tarjeta de crédito. A partir del momento de registro la *FPN* hace cargos mensuales a la cuenta o tarjeta de crédito del donante, entregando a cambio 0.2 puntos por cada $1.00 peso (moneda nacional mxn) donado.

Si el donador se registra con tarjeta de crédito, la información de la misma se almacena en el sistema de forma inmediata y transparente al usuario, asimismo ésta queda resguardada para los operadores técnicos del programa; es decir, sólo es accesible a través de una contraseña que únicamente el director de la *FPN* conoce.

Al momento de registrarse, el donador es colocado dentro de una estructura a la que los operadores técnicos del programa llaman pirámide, si algún donador invita a más donantes, los invitados pueden a su vez invitar a otros, estos últimos a otros más y así sucesivamente.

Las pirámides se pueden modelar usando un árbol binario en donde cada donador es la raíz de su propio subárbol. Por ejemplo, si el donador `A` invita al donador `B` al programa, el donador `B` queda registrado en el primer espacio disponible recorriendo el subárbol del donador `A` de arriba hacia abajo y de izquierda a derecha.

Mensualmente cada donador recibe una cantidad adicional de diez puntos por cada 2^n (con $n >= 1$) donadores registrados en el nivel n de su subárbol, esto es, por cada nivel completo.

Ilustraremos el funcionamiento del programa con un ejemplo. Supongamos el siguiente escenario:

1. El donador `A` se registra en una nueva pirámide e invita a los donantes `B` y `C`.
2. El donador `B` invita al `D` y el donador `C` invita al `E`.
3. El donador `A` invita al `F`.
4. El donador `D` invita al `G`.
5. El donador `A` invita al `H`.

Siguiendo la estrategia mencionada, la construcción de la pirámide final de ocho donadores se llevaría a cabo de la manera ilustrada en la Figura 1.1.

Si un donador quisiera participar varias pirámides a la vez, tendría que registrarse con datos bancarios y correo electrónico distintos.

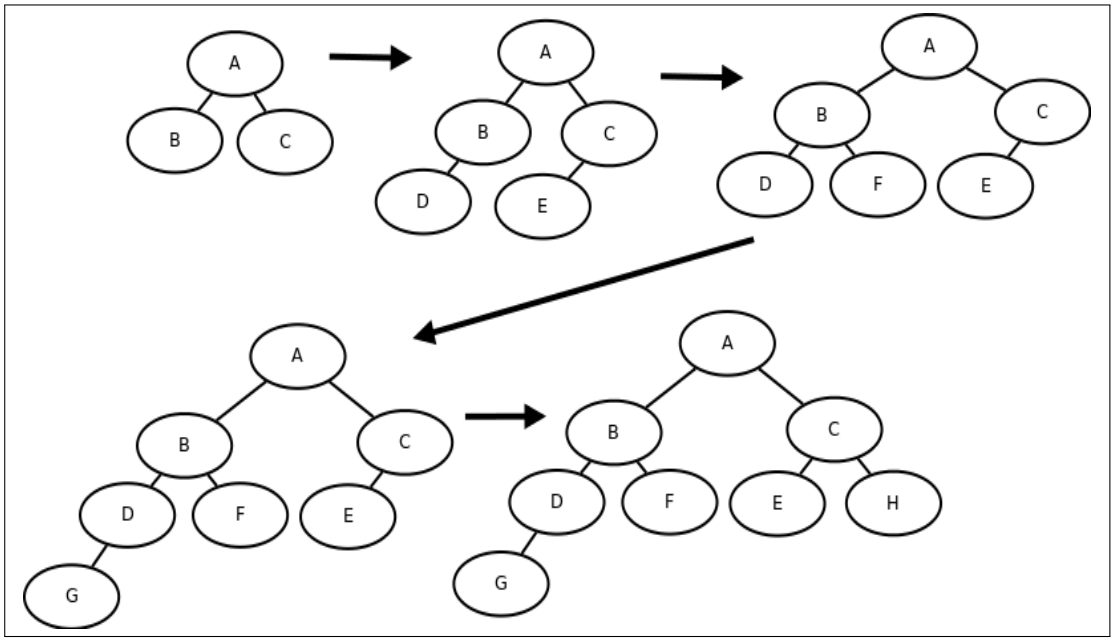


Figura 1.1: Ejemplo de pirámide

En este caso, el donador `A` recibiría veinte puntos adicionales al mes puesto que dentro de su subárbol el nivel 1 tiene dos invitados y el nivel 2 tiene cuatro invitados. Por otro lado, los donadores `B` y `C` recibirían solamente diez puntos dado que cuentan con dos invitados en el primer nivel de sus respectivos subárboles.

Se ha establecido un catálogo de los artículos almacenados por la *Fundación* que un donador puede intercambiar por sus puntos acumulados. Cada artículo tiene un valor en puntos establecido directamente por la *FPN*.

Todos los puntos adquiridos por un donador tienen vigencia de un año a partir de la fecha en la que se consiguieron.

1.2.2 Trabajo previo

Con el objetivo de poner en marcha el programa *Coleccionando Sonrisas* la *FPN* contrató a una empresa de desarrollo de software para que implementara un sistema que se encuentra actualmente en funcionamiento. Este sistema permite realizar tareas básicas de gestión sobre el catálogo de bienes almacenados en la bodega de la *Fundación*.

El sistema mencionado fue desarrollado con HTML, Javascript, PHP y existía una base de datos relacional implementada en el manejador MySQL. Cabe destacar que hasta este punto el sistema solamente cubre las necesidades relacionadas con los artículos (a los que denominan productos) intercambiables por puntos. Se pueden distinguir las siguientes funciones en el mismo:

- Almacenamiento de información relacionada con productos.
- Registro de nuevos productos.
- Consulta de catálogo de productos.
- Edición de información de productos previamente almacenados.

Describiremos brevemente cada una de estas funciones a continuación.

Almacenamiento de información de productos

El sistema permite almacenar la información relacionada con los productos, bienes y artículos que se encuentran almacenados en la bodega de la *FPN*. Para cada producto están disponibles los siguientes datos:

- Identificador unívoco en la base de datos.
- Nombre.
- Categoría: entretenimiento, deportes, tecnología, hogar, cultura, sonrisas o vivenciales.
- Descripción.
- Cantidad de artículos del producto almacenados en la bodega.
- Valor de mercado en pesos mexicanos.
- Valor en puntos.
- Demanda esperada por parte de los donadores: muy baja, baja, promedio, alta o muy alta.

- Representación gráfica del producto (dibujo, fotografía, diagrama, etcétera).
- Fecha de ingreso del producto a la base de datos.

Registro de nuevos productos

Para registrar un nuevo producto los operadores técnicos del programa *Coleccionando Sonrisas* llenan un formulario como el que se muestra en la Figura 1.2

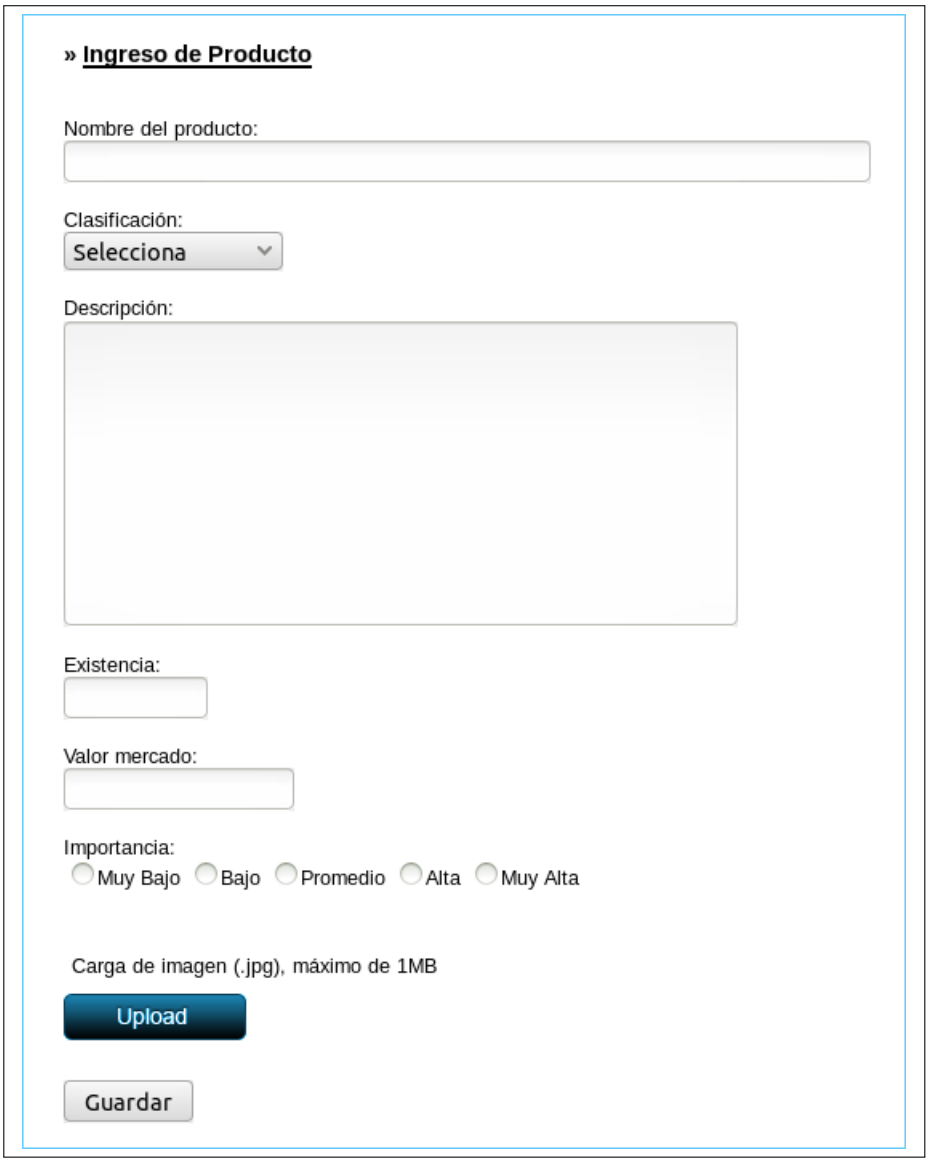
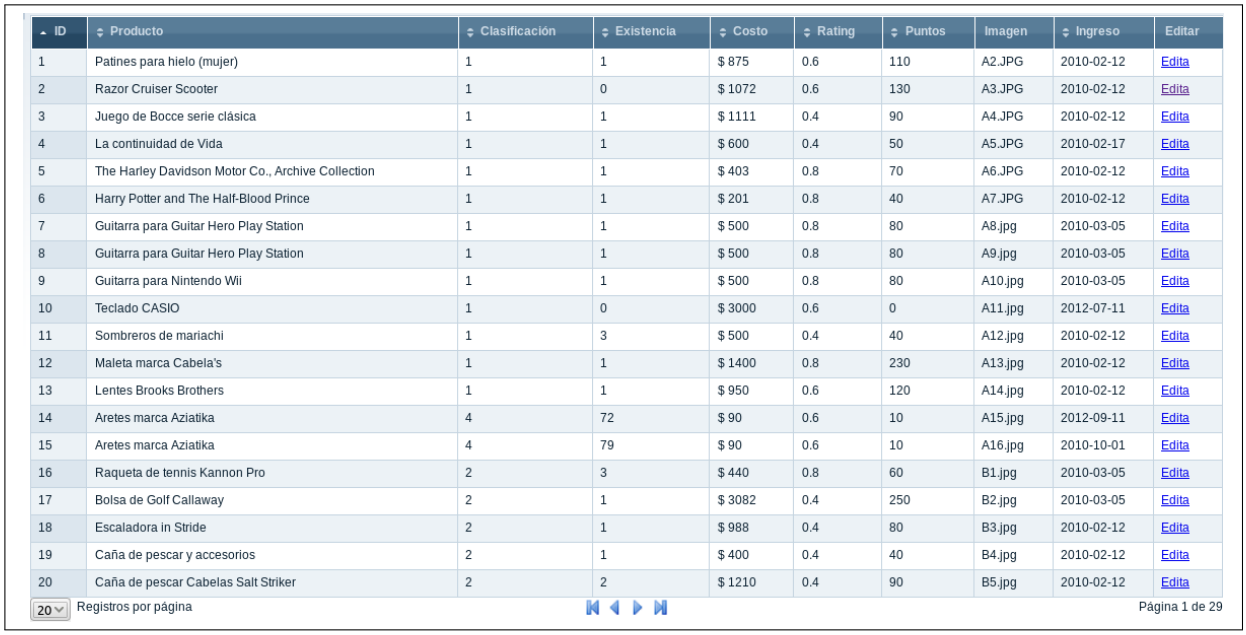


Figura 1.2: Formulario de registro de productos

Consulta del catálogo de productos

El catálogo de productos muestra de manera amigable al usuario una lista completa de productos disponibles. En cada renglón de la tabla hay una liga que lleva a la página de edición para el producto en cuestión. En la Figura 1.3 se muestra la vista correspondiente en la que aparecen en cada renglón un producto distinto, las columnas representan sus atributos, que son: id, producto(nombre), clasificación, existencia, costo, *rating*, puntos, imagen, ingreso y editar.



ID	Producto	Clasificación	Existencia	Costo	Rating	Puntos	Imagen	Ingreso	Editar
1	Patines para hielo (mujer)	1	1	$ 875	0.6	110	A2.JPG	2010-02-12	Edita
2	Razor Cruiser Scooter	1	0	$ 1072	0.6	130	A3.JPG	2010-02-12	Edita
3	Juego de Bocce serie clásica	1	1	$ 1111	0.4	90	A4.JPG	2010-02-12	Edita
4	La continuidad de Vida	1	1	$ 600	0.4	50	A5.JPG	2010-02-17	Edita
5	The Harley Davidson Motor Co., Archive Collection	1	1	$ 403	0.8	70	A6.JPG	2010-02-12	Edita
6	Harry Potter and The Half-Blood Prince	1	1	$ 201	0.8	40	A7.JPG	2010-02-12	Edita
7	Guitarra para Guitar Hero Play Station	1	1	$ 500	0.8	80	A8.jpg	2010-03-05	Edita
8	Guitarra para Guitar Hero Play Station	1	1	$ 500	0.8	80	A9.jpg	2010-03-05	Edita
9	Guitarra para Nintendo Wii	1	1	$ 500	0.8	80	A10.jpg	2010-03-05	Edita
10	Teclado CASIO	1	0	$ 3000	0.6	0	A11.jpg	2012-07-11	Edita
11	Sombreros de mariachi	1	3	$ 500	0.4	40	A12.jpg	2010-02-12	Edita
12	Maleta marca Cabela's	1	1	$ 1400	0.8	230	A13.jpg	2010-02-12	Edita
13	Lentes Brooks Brothers	1	1	$ 950	0.6	120	A14.jpg	2010-02-12	Edita
14	Aretes marca Aziatika	4	72	$ 90	0.6	10	A15.jpg	2012-09-11	Edita
15	Aretes marca Aziatika	4	79	$ 90	0.6	10	A16.jpg	2010-10-01	Edita
16	Raqueta de tennis Kannon Pro	2	3	$ 440	0.8	60	B1.jpg	2010-03-05	Edita
17	Bolsa de Golf Callaway	2	1	$ 3082	0.4	250	B2.jpg	2010-03-05	Edita
18	Escaladora in Stride	2	1	$ 988	0.4	80	B3.jpg	2010-02-12	Edita
19	Caña de pescar y accesorios	2	1	$ 400	0.4	40	B4.jpg	2010-02-12	Edita
20	Caña de pescar Cabelas Salt Striker	2	2	$ 1210	0.4	90	B5.jpg	2010-02-12	Edita

20 Registros por página

Página 1 de 29

Figura 1.3: Vista de catálogo de productos

Edición de productos

A través de esta herramienta los operadores técnicos del programa *Coleccionando Sonrisas* pueden cambiar la información relacionada con un producto. Aunque esencialmente se pretende que sólo se actualice la cantidad almacenada en la bodega[3], también permite cambiar la representación gráfica, el nombre y todos los campos capturados inicialmente a través de la interfaz mostrada en la Figura 1.2 en la que la forma aparecerá previamente llenada con los datos actuales del producto a modificar.

[3]Por ejemplo, la cantidad de teléfonos celulares Marca *X*, modelo *X213*

1.3 Requerimientos generales

El objetivo general que se persigue al incluir al sustentante en el proceso de desarrollo es extender el sistema ya implementado para que funcione como una herramienta en línea a través de la cual se pueda gestionar el programa *Coleccionando Sonrisas* de modo tal que los donadores puedan registrarse, revisar, consultar e intercambiar sus puntos acumulados a través del web, con plena confianza en la preservación de la confidencialidad de sus datos personales y bancarios (por un lado), y por el otro aportar a la *FPN* una herramienta sólida con la que los operadores técnicos del programa *Coleccionando Sonrisas* puedan realizar la mayor cantidad posible de las tareas relacionadas con el mismo.

Podríamos pensar en el proceso de desarrollo en el que participará el sustentante como un proyecto de diseño e implementación de software en el que los requerimientos funcionales son los objetivos generales descritos en el párrafo precedente.

A grandes rasgos, es posible distinguir entre tres objetivos particulares de este trabajo:

- Desarrollar herramientas tecnológicas para donadores.
- Desarrollar herramientas tecnológicas para operadores técnicos.
- Desarrollar software de automatización de procesos.

1.3.1 Herramientas para donadores

Como ya se ha mencionado, un donador es aquel que participa en el programa aportando cierta cantidad de dinero y acumulando puntos que puede intercambiar posteriormente por productos dentro del catálogo.

A este respecto será necesario que el sistema aporte herramientas a los donadores para:

- Registrarse en el programa.
- Consultar sus puntos acumulados.
- Solicitar el intercambio de puntos acumulados por beneficios.
- Consultar el catálogo de beneficios.
- Consultar los detalles de cada beneficio (valor en puntos, descripción, etcétera).

1.3.2 Herramientas para operadores técnicos del programa

Para evitar que los registros relacionados con el programa *Coleccionando Sonrisas* se lleven en papel, el sistema tendrá una incidencia fundamental en el mejoramiento de la eficiencia de los operadores técnicos en el desempeño de de las siguientes tareas:

- Consulta de datos de los donadores.
- Cobranza mensual[4].
- Adición y sustracción manual de puntos vigentes.

1.3.3 Automatización de procesos

Entenderemos que un proceso está automatizado cuando puede llevarse a cabo exitosamente sin la intervención (o con mínima intervención) de un operador técnico del programa *Coleccionando Sonrisas*.

Los procesos que se pretende automatizar son:

- Abono mensual de puntos obtenidos por donativo, esto es, los que se reciben a cambio de la confirmación de un cargo a tarjeta de crédito, tarjeta de débito o CLABE.
- Abono mensual de puntos adicionales por donadores invitados respetando las reglas de las pirámides (árboles binarios).
- Invalidación de puntos caducos, es decir, después de un año de haberse abonado a la cuenta del usuario los puntos que no se intercambiaron deberán ser marcados como no válidos, esto es, ya no podrán trocarse por productos.

[4]La *FPN* tiene un convenio con una institución bancaria para usar la infraestructura de *web-services Payworks®* con el fin de facilitar los cargos a tarjetas de crédito, débito y clave bancaria estandarizada (CLABE).

Capítulo 2

Especificación

En este capítulo se describe con detalle el sistema, en términos de los requerimientos de la *FPN*. Considerando el alcance del proyecto de desarrollo, se dividirá la especificación en cuatro rubros principales:

- Roles de usuario.
- Casos de uso generales.
- Casos de uso administrativos.
- Procesos asistidos.

2.1 Roles de usuario

En adelante entenderemos como usuario a toda persona que aprovecha las funciones del sistema para realizar alguna actividad. Dicho aprovechamiento se puede llevar a cabo de dos maneras distintas: a través del portal web del programa *Coleccionando Sonrisas* o a través del administrador del sistema. Si bien se plantea al administrador del sistema como un recurso para los demás usuarios, también es cierto que éste es un usuario en sí mismo, sin embargo, puede realizar sus funciones a través de herramientas "*menos amigables*"[1].

A continuación se describen todos los roles contemplados en el sistema con sus respectivos privilegios.

[1]Intérprete de comandos, acceso directo al manejador de base de datos, etcétera.

2.1.1 Visitante

Un visitante es cualquier persona que observa una de las páginas del programa en el navegador web de algún dispositivo con acceso a Internet. Los visitantes pueden acceder a cualquiera de las pantallas públicas que presenta el sistema a través de su portal.

2.1.2 Donador

Un donador es un visitante que en algún momento capturó sus datos a través de alguno de los formularios[2] destinados para este efecto. Después de completar su registro, los usuarios con este rol tendrán acceso a la página de perfil en donde podrán consultar los puntos que tienen acumulados, la cantidad de donadores que están registrados y el número de niveles completos en su subárbol, etcétera. Asimismo podrán utilizar sin restricciones la herramienta para intercambiar puntos por productos.

2.1.3 Operador técnico

Este rol está reservado para empleados o voluntarios de la *FPN*, quienes pueden llevar a cabo tareas a través de las páginas correspondientes como:

1. Registrar nuevos productos.
2. Editar productos registrados previamente.
3. Consultar el catálogo de productos.
4. Abonar o restar puntos a un donador específico.
5. Consultar cuántos niveles completos hay en el subárbol de un donador específico.
6. Crear nuevas pirámides (árboles).
7. Registrar conjuntos de donadores a través de una hoja de cálculo[3].
8. Generar archivos de cargos mensuales a tarjetas de crédito, esto debe llevarse a cabo bajo la supervisión del director de la *FPN*.

[2] Se describirán con detalle estos formularios más adelante, en los casos de uso.
[3] En particular se utiliza Microsoft® Excel®.

Todo usuario con este rol deberá ser registrado manualmente en el sistema por el administrador.

2.1.4 Supervisor

En los casos en que un operador técnico necesite tener acceso a información sensible almacenada dentro de la base de datos, será necesario que un usuario adicional se encuentre presente, dicho usuario deberá estar registrado con el rol de supervisor.

2.1.5 Administrador

Este rol permitirá al usuario ejecutar programas basados en línea de comandos para:

- Generar archivos de intercambio de información que se enviarán a *Payworks®*.
- Analizar archivos de intercambio de información generados por *Payworks®*.
- Llevar a cabo la distribución mensual de puntos, tanto normales como adicionales.

Por lo que todo usuario administrador deberá tener conocimientos amplios sobre *Bash*[4] y también sobre el lenguaje de programación usado para desarrollar el sistema.

2.2 Casos de uso generales

A continuación se presentan los casos de uso del sistema que involucran a usuarios cuyo rol es de visitante o donador.

2.2.1 Registro de donador

El usuario visitante tendrá acceso a una pantalla en la que podrá elegir el modo en el que desea registrarse. Dependiendo de su elección se presentará un formulario en el que capturará todos sus datos personales y los correspondientes a su forma de pago. Dependiendo de la forma de pago elegida dará inicio un procedimiento específico que puede requerir la participación del operador técnico y el administrador.

[4]Bash: *Bourne-again shell* es el intérprete de comandos predeterminado para la mayoría de las distribuciones del sistema operativo GNU/Linux[18]

Además de sus datos personales y bancarios, el usuario podrá registrar un nombre de referencia, esto es, el nombre de usuario de algún donante registrado o bien la palabra clave de alguna de las pirámides. Este campo será crucial en el proceso de registro del nuevo donador, si se ha capturado un nombre de usuario el nuevo registro se ubicará en el primer lugar disponible dentro del subárbol del que el referido es raíz. En caso de haber capturado la palabra clave de una pirámide el nuevo registro se creará en el árbol correspondiente a la misma. En cualquier otro caso se ubicará al nuevo donante en la pirámide general.

Tarjeta de débito o CLABE

Para este modo de registro es necesario llevar a cabo un proceso no automatizado cuyas etapas específicas se describen en los siguientes apartados. Solamente podrá registrarse una de las opciones dentro del sistema, esto es, un mismo donante no podrá registrarse con su clave bancaria estandarizada y después con el número de su tarjeta de débito. Por otro lado, las políticas de seguridad del sistema *Payworksr®* impiden registrar a dos donadores con los mismos datos bancarios, esta limitante también se implementará en el sistema *Coleccionando Sonrisas*. El diagrama de caso de uso se muestra en la Figura 2.1.

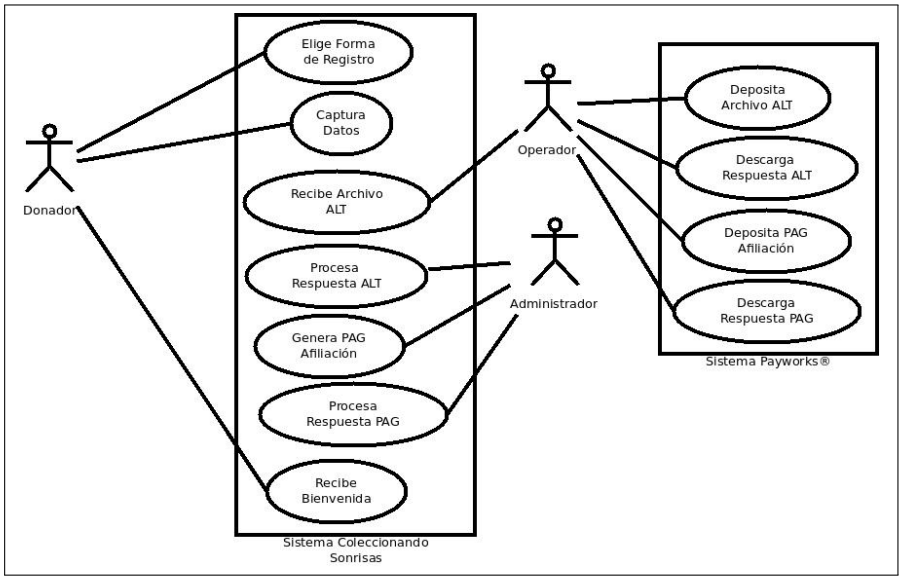


Figura 2.1: Registro por CLABE o tarjeta de débito

Registro del usuario

Al enviar los datos, el usuario será dirigido a una pantalla en la que se le notifica que ha iniciado un proceso de afiliación que podrá durar entre de 24 a 48[5]. Al mismo tiempo, el sistema generará un correo electrónico dirigido al operador técnico registrado que llevará como adjunto un archivo de afiliación de cuentas *Payworksr®*, en adelante ALT.

Proceso de archivo en *Payworks®*

Tras recibir el archivo mencionado en el párrafo anterior, el operador técnico realizará un proceso a través del portal de *Payworks®* que generará un archivo de respuesta.

Respuesta de afiliación *Payworks®*

El sistema *Payworks®* genera un archivo de respuesta en un formato específico que se debe analizar para saber si la afiliación solicitada se completó con éxito o no. El operador técnico debe hacer llegar al administrador el archivo de respuesta en cuestión para que éste pueda revisarlo y notificar la causa del rechazo de la afiliación o bien, continuar con el proceso de registro de usuario descrito en el siguiente párrafo.

Cobranza por afiliación

Cuando se completa exitosamente una afiliación a *Payworks®* el administrador debe generar un archivo de requerimiento de pago (en adelante PAG) con el que se realiza el primer cargo a la cuenta del donador y enviárselo al operador técnico inmediatamente.

[5]Tiempo estimado que se toma procesar el archivo de afiliación de cuentas y la cobranza por afiliación.

Respuesta cobranza *Payworks®*

Al igual que los archivos ALT, los PAG deben ser procesados a través del portal *Payworks®*, tras lo cual se generará una respuesta que deberá recibir el administrador para su análisis. En caso de que no se haya podido realizar el cargo a la cuenta del donante, el administrador deberá reportar el motivo del rechazo. De lo contrario se deberá completar el registro del donador en el sistema *Coleccinando Sonrisas* y hacer los ajustes necesarios para que el nuevo donante pueda visitar la página de su perfil de usuario, intercambiar sus puntos por productos, etcétera.

Tarjeta de crédito

A diferencia del modo anterior el registro con tarjeta de crédito se puede resolver inmediatamente, esto es, el nuevo donador no necesita esperar para poder usar su cuenta dentro del sistema. Además, al ser una plataforma flexible, será posible que varios usuarios donen a través del mismo plástico.

Al completar la captura de sus datos y enviar el formulario el sistema genera una petición `HTTPS` al *web-service* correspondiente de *Payworks®*. Dependiendo del valor con el que dicho servicio responda se dirigirá al usuario a la pantalla de registro exitoso o a una pantalla de notificación de error de pago. La Figura 2.2 representa el diagrama de este caso de uso.

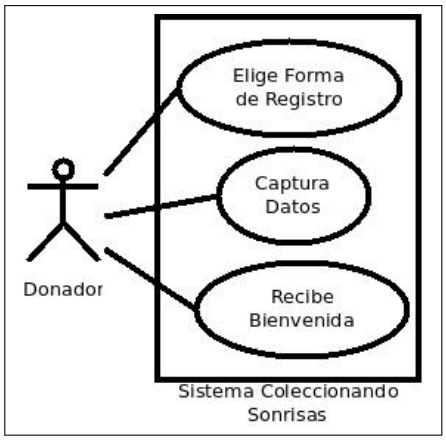


Figura 2.2: Registro por tarjeta de crédito

2.2.2 Catálogo de productos detallado

Los visitantes podrán consultar un catálogo en el que se muestran todos los productos agrupados en las siguientes categorías:

1. Vivenciales.
2. Sonrisas.
3. Entretenimiento.
4. Deportes.
5. Tecnología.
6. Hogar.
7. Cultura.
8. Todos.

En dicho catálogo se mostrará una liga por cada una de las categorías mencionadas. Al seguir dicha liga se recargará la pantalla mostrando sólo los productos de la categoría elegida. Para cada producto se mostrará una fotografía o una liga a los detalles del mismo.

La pantalla aquí descrita contendrá también referencias hacia los siguientes módulos:

- Pantalla principal.
- Formularios de registro.
- Pantalla de inicio de sesión.

También estará disponible una caja de selección donde el usuario podrá elegir el rango de valores de los productos que desea consultar. Al elegir una opción se filtrarán los elementos correspondientes a los productos cuyo valor no esté dentro del rango seleccionado.

2.2.3 Detalles de producto

Se mostrarán todos los datos acerca del producto almacenados en la base de datos. Los botones mostrados dependerán del estado de la sesión del usuario.

Si la sesión está cerrada se mostrarán los siguientes botones:

Inscríbete: Al seleccionarlo se abrirá la página de registro de usuarios.

Iniciar Sesión: Al seleccionarlo se abrirá la página de inicio de sesión.

Regresar: Al seleccionarlo se abrirá la página principal del sistema.

Adquirir: Este botón aparecerá deshabilitado pues no se pueden adquirir productos sin abrir una sesión.

Si se detecta una sesión abierta se presentarán los siguientes botones:

Tu cuenta: Al seleccionarlo se abrirá la página de perfil del usuario.

Cerrar Sesión: Al seleccionarlo se abrirá la página principal y se cerrará la sesión del usuario en el sistema.

Regresar: Al seleccionarlo se abrirá la página principal del sistema.

Adquirir: Al seleccionar este botón se iniciará el proceso de adquisición de producto para el usuario y producto seleccionado.

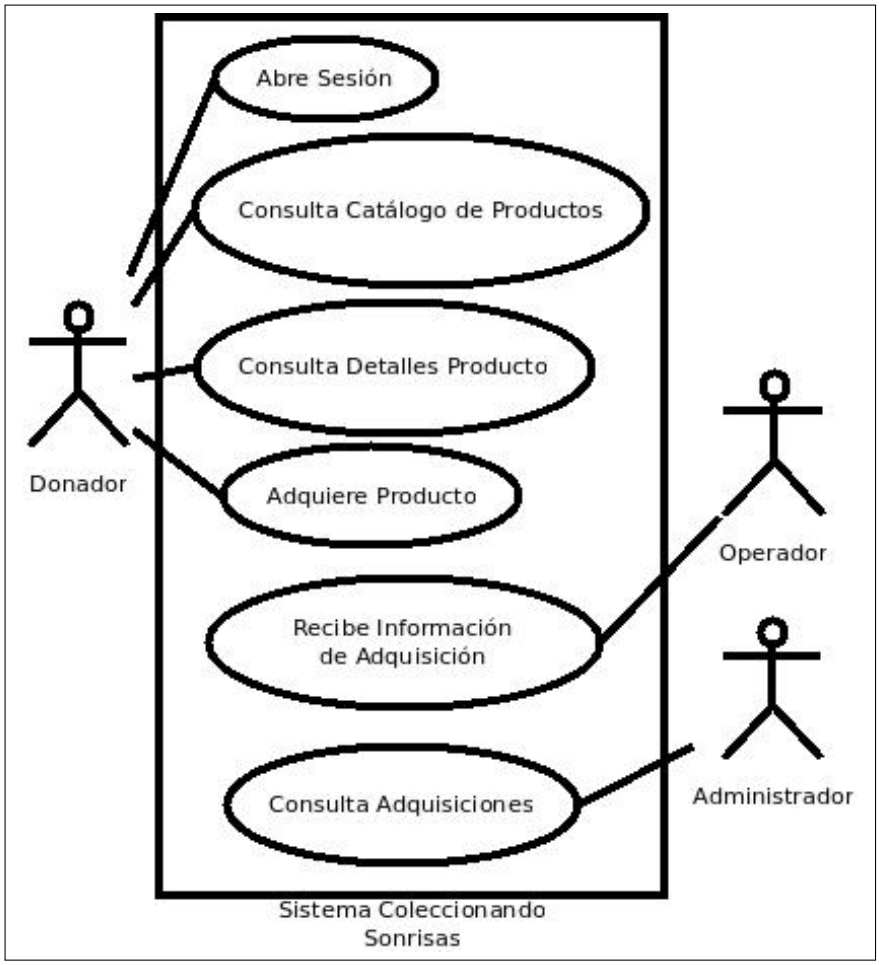


Figura 2.3: Intercambio de puntos por productos

Proceso de adquisición

Tras elegir un producto usando el botón adquirir se restará el valor en puntos de los acumulados en la cuenta del usuario comenzando por los que tienen mayor antigüedad. Tras restar los puntos de la cuenta del usuario se reducirá en uno el número de elementos del producto elegido registrados en el inventario y se generará un correo electrónico dirigido al operador técnico describiendo los detalles de la adquisición. Esta información quedará almacenada en la base de datos para consulta del administrador. Se presenta el diagrama de este caso de uso en la Figura 2.3

2.2.4 Pantalla de perfil de usuario

En esta pantalla, un donador podrá consultar toda la información relacionada con su cuenta, además de los datos que se tienen almacenados sobre el usuario, en esta pantalla se mostrarán los siguientes:

1. Información calculada al momento de la visita.
2. Imagen de la pirámide en la que está registrado el donador.
3. Mensaje de bienvenida de la pirámide en la que está registrado el donante.

Se describe en las secciones subsecuentes los datos del primer punto y la manera en la que serán calculados.

Puntos del mes

Este valor representa a la cantidad total de puntos que el donante recibirá en el próximo corte mensual tomando en cuenta el estado de su subárbol en el momento de la consulta. Este número se calculará sumando 50 (que es la cantidad recibida al confirmar el pago de la cuota mensual) más $10 * n$ donde n es el número de niveles completos que existen en el subárbol del que el donante en cuestión es raíz.

Niveles debajo

Es el número de niveles completos dentro de subárbol del que el donante es raíz. Como se ha mencionado anteriormente, se considera un nivel como completo si cuenta con 2^n invitados registrados donde n es la distancia entre ellos y la raíz. Es

importante notar que se consideran todos los usuarios registrados no importando su estado (baja, suspensión, etcétera.).

Amigos en tu pirámide

Es el número de donantes registrados en el subárbol del que el usuario en cuestión es la raíz. Se considera en esta cuenta a todos los donantes registrados en el subárbol, esto es, se suman a la cuenta todos los usuarios que están dados de baja o suspendidos al momento de la consulta.

2.2.5 Formulario para juntar puntos

Dentro del perfil del usuario existirá una opción a través de la cual un donador pueda regalar cierta cantidad de puntos a otro. El sistema deberá recabar el nombre de usuario del destinatario y la cantidad de puntos a obsequiar (en múltiplos de 10 entre 10 y 100) verificando la existencia del receptor y que la cantidad de puntos del remitente sea mayor a la que éste eligió.

De cumplir con los requisitos establecidos, se restarán los puntos de la cuenta del donante y se agregarán a la del destinatario. Además, se enviará un correo electrónico al receptor. En otro caso se deberá mostrar un mensaje explicando el motivo por el que no se puede llevar a cabo la transacción.

2.3 Casos de uso administrativos

En esta sección se describirán con detalle los casos de uso relacionados con los usuarios que tienen el rol de operador técnico. Ningún usuario visitante o donador tendrá autorización para llevar a cabo estas actividades.

2.3.1 Creación de pirámides

Para crear una pirámide nueva es necesario proveer los siguientes datos a través del formulario correspondiente:

Mensaje de bienvenida: El texto que se desplegará en el perfil de los usuarios registrados en esta pirámide.

Usuario clave: Palabra clave que podrá ser usada en el formulario de registro de usuarios para ubicarlos en esta pirámide.

Imagen: Imagen a mostrar en el perfil de los usuarios registrados en esta pirámide.

Al enviar los datos capturados el sistema creará los registros correspondientes de modo tal que los visitantes podrán registrarse en la nueva pirámide de manera inmediata.

2.3.2 Adición y sustracción de puntos

Dado que los usuarios pueden adquirir puntos adicionales haciendo uso de métodos que no están soportados por el sistema descrito en este texto, se ofrecerán herramientas para que el operador técnico pueda cambiar de manera manual el número de puntos acumulados por un donante.

Si se agregan puntos quedará registrada la fecha en la que se realizó el abono, en caso de sustracción se restará la cantidad de puntos comenzando por los de mayor antigüedad.

2.3.3 Cálculo de puntos por usuario

Existirá una herramienta para que el operador técnico pueda calcular la cantidad de puntos que un usuario dado recibirá en el próximo corte mensual. La estrategia para el cálculo de esta cantidad será la descrita en la Sección 2.2.4 y será necesario proveer el identificador del usuario.

2.3.4 Generación de archivos de cobranza

Los operadores técnicos, bajo la estricta supervisión de un representante del patronato de la fundación podrán generar un archivo en formato de hoja de cálculo requerido por el sistema *Payworks@* para procesar los cargos mensuales a tarjetas de crédito de los donadores. Deberá existir una herramienta que permita esto garantizando que el operador técnico no podrá tener acceso esta información sin la autorización y vigilancia de la *FPN*.

2.3.5 Inventario de productos

El operador técnico tendrá acceso a una pantalla en la que se listarán todos los productos registrados en el sistema con sus respectivos detalles a manera de tabla. A continuación se muestra el diseño gráfico de la pantalla donde se puede apreciar que cada producto está representado por un renglón y sus respectivos atributos se organizan en las columnas cuyos encabezados son: id, producto(nombre), clasificación, existencia, costo, *rating*, puntos, imagen, ingreso y editar. Adicionalmente a sus atributos, se incluye una liga a través de la cual se podrá acceder a la pantalla de edición de dicho producto. La información que se despliega en esta tabla se describirá con detalle en la Sección 3.1.7.

ID	Producto	Clasificación	Existencia	Costo	Rating	Puntos	Imagen	Ingreso	Editar
1	Patines para hielo (mujer)	1	1	$ 875	0.6	110	A2.JPG	2010-02-12	Edita
2	Razor Cruiser Scooter	1	0	$ 1072	0.6	130	A3.JPG	2010-02-12	Edita
3	Juego de Bocce serie clásica	1	1	$ 1111	0.4	90	A4.JPG	2010-02-12	Edita
4	La continuidad de Vida	1	1	$ 600	0.4	50	A5.JPG	2010-02-17	Edita
5	The Harley Davidson Motor Co., Archive Collection	1	1	$ 403	0.8	70	A6.JPG	2010-02-12	Edita
6	Harry Potter and The Half-Blood Prince	1	1	$ 201	0.8	40	A7.JPG	2010-02-12	Edita
7	Guitarra para Guitar Hero Play Station	1	1	$ 500	0.8	80	A8.jpg	2010-03-05	Edita
8	Guitarra para Guitar Hero Play Station	1	1	$ 500	0.8	80	A9.jpg	2010-03-05	Edita
9	Guitarra para Nintendo Wii	1	1	$ 500	0.8	80	A10.jpg	2010-03-05	Edita
10	Teclado CASIO	1	0	$ 3000	0.6	0	A11.jpg	2012-07-11	Edita
11	Sombreros de mariachi	1	3	$ 500	0.4	40	A12.jpg	2010-02-12	Edita
12	Maleta marca Cabela's	1	1	$ 1400	0.8	230	A13.jpg	2010-02-12	Edita
13	Lentes Brooks Brothers	1	1	$ 950	0.6	120	A14.jpg	2010-02-12	Edita
14	Aretes marca Aziatika	4	72	$ 90	0.6	10	A15.jpg	2012-09-11	Edita
15	Aretes marca Aziatika	4	79	$ 90	0.6	10	A16.jpg	2010-10-01	Edita
16	Raqueta de tennis Kannon Pro	2	3	$ 440	0.8	60	B1.jpg	2010-03-05	Edita
17	Bolsa de Golf Callaway	2	1	$ 3082	0.4	250	B2.jpg	2010-03-05	Edita
18	Escaladora in Stride	2	1	$ 988	0.4	80	B3.jpg	2010-02-12	Edita
19	Caña de pescar y accesorios	2	1	$ 400	0.4	40	B4.jpg	2010-02-12	Edita
20	Caña de pescar Cabelas Salt Striker	2	2	$ 1210	0.4	90	B5.jpg	2010-02-12	Edita

20 Registros por página Página 1 de 29

Figura 2.4: Inventario de productos

La tabla podrá ordenarse de manera ascendente o descendente usando los valores de cada una de las columnas. En cada renglón aparecerá una liga a través de la cual el operador técnico podrá cambiar los datos del producto seleccionado.

2.3.6 Creación y edición de productos

El operador técnico podrá cambiar los datos registrados acerca de cualquier producto en particular o registrar productos nuevos a través del mismo formulario. El formulario mostrado en la Figura 2.5 aparecerá previamente llenado cuando se trate de edición de productos.

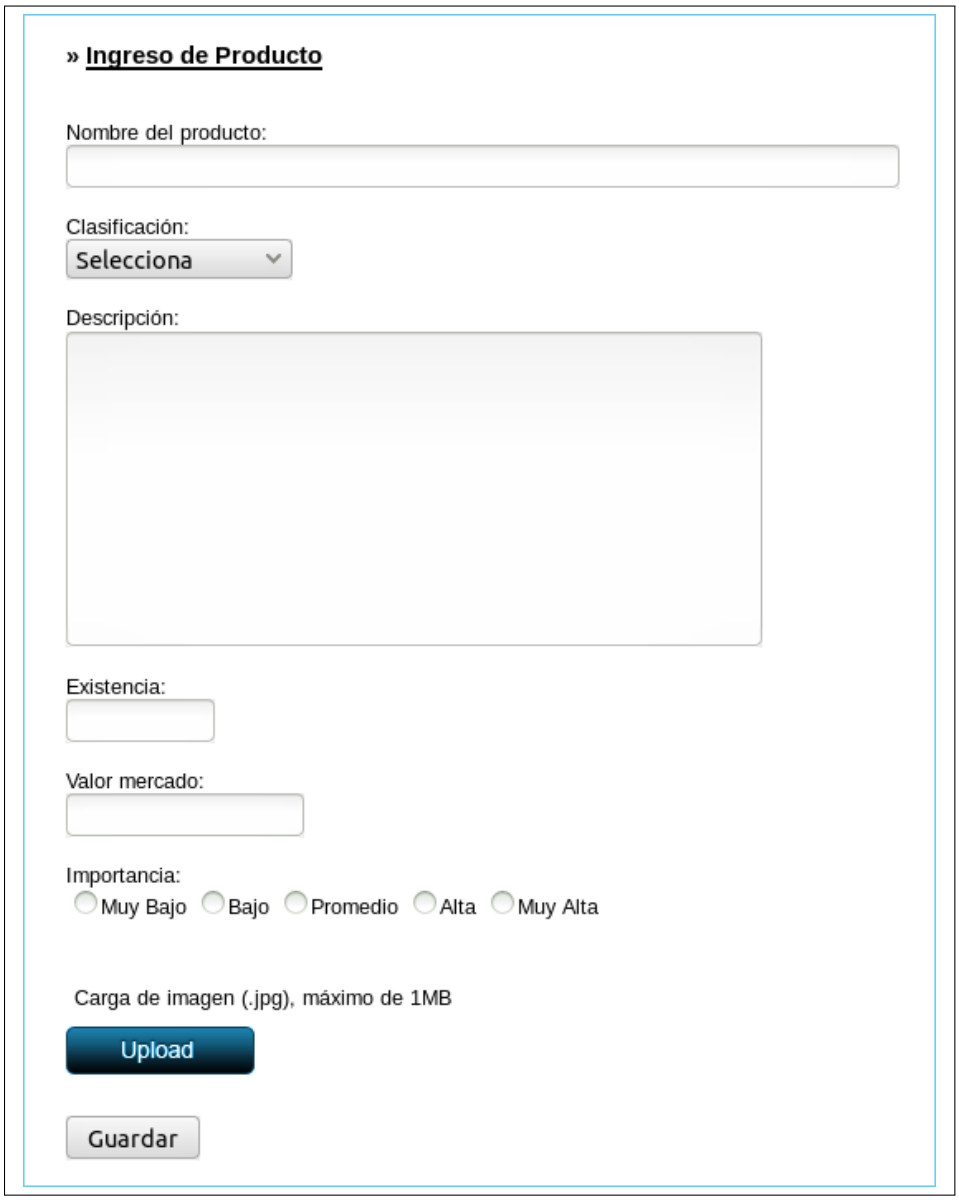


Figura 2.5: Formulario de edición o registro de productos

2.4 Procesos asistidos

Entenderemos que un proceso es asistido si requiere la intervención de un usuario con rol de administrador. Al no contar con el tiempo y presupuesto suficientes para automatizar la totalidad de estos procesos, la *FPN* tomó la decisión de no implementar módulos e interfaces de usuario que permitieran a los operadores técnicos realizar estas funciones reduciendo así los costos y el tiempo total que tomaría el desarrollo del proyecto. Los procesos asistidos que se contemplan dentro del presente trabajo son:

- Distribución mensual de puntos.
- Proceso de archivos de intercambio de información.

2.4.1 Distribución mensual de puntos

Este proceso comenzará con la solicitud explícita de algún operador técnico al administrador. Tras recibir la petición, éste ejecutará en un principio un programa basado en línea de comandos que generará una representación gráfica para cada pirámide dentro de la base de datos. Estos archivos se publicarán para que los operadores técnicos puedan consultarlos y en ellos deberán aparecer los datos de cada donante así como la cantidad de puntos que tuvieran acumulada al momento de la ejecución.

Una vez publicadas las gráficas iniciales, se deberá ejecutar otro programa de línea de comandos que recibirá como parámetro la ruta de un archivo de texto plano con los identificadores de los usuarios a quienes se debe excluir, esto es, a quienes no se deberá abonar puntos.

Habiendo ejecutado exitosamente el programa en cuestión, el administrador procederá a generar de nuevo las representaciones gráficas de las pirámides con el fin de que los operadores técnicos puedan consultar el las nuevas cantidades de puntos que han acumulado los usuarios y así verificar la aplicación correcta de las reglas de distribución[6]. En la Figura 2.6 se puede apreciar la representación gráfica de una pirámide completa de tres niveles de altura.

[6]Estas reglas están descritas en la Sección 1.2.1.

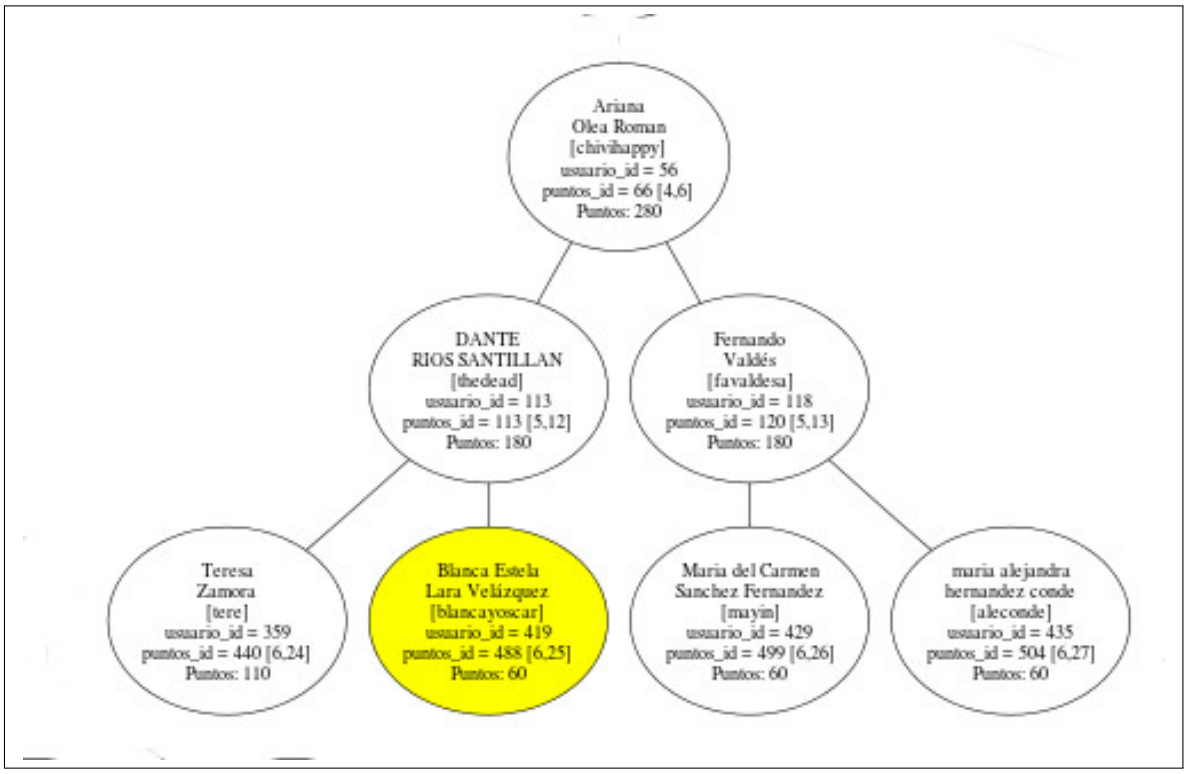


Figura 2.6: Representación gráfica de pirámide

2.4.2 Proceso de archivos de intercambio

La plataforma *Payworks®* se comunicará con el sistema a través de archivos de texto plano en formato de longitud fija[7] con una línea de encabezado. La especificación de estos archivos se encuentra debidamente documentada en los manuales de *Payworks®* cuyos detalles no es posible revelar en este documento sin violar el convenio de privacidad firmado entre la *FPN* y el proveedor del servicio.

El sistema podrá generar dos tipos de archivos, a saber, cobranza y afiliación, a través de los respectivos programas basados en línea de comandos. Estos archivos se generarán con la información almacenada en la base de datos y deberán cumplir con todas las características y estándares definidos en los manuales de *Payworks®*

[7]Traducción literal de *fixed-width format* que describe archivos en los que cada registro está representado por una línea y sus campos tienen una longitud específica.

Capítulo 3

Base de Datos

Abordaremos en este capítulo el diseño de la base de datos de la aplicación detallando las entidades, normalización, referencias y seguridad en capa de datos. Partiendo de la descripción de cada uno de los agentes relacionados con el sistema y sus respectivos atributos llegaremos al diseño de la base de datos, esto es, a la representación de cada uno de los agentes ya mencionados en el modelo Entidad-Relación[1].

3.1 Modelo de Entidades

Hablaremos de cada una de las entidades a considerar en el proyecto describiendo los atributos que las caracterizarán. Es importante destacar que la intención de esta sección es solamente plantear el panorama de los agentes participantes en el sistema, la representación tabular se abordará en las secciones subsecuentes.

3.1.1 Usuario

Con esta entidad se representan las personas involucradas con el sistema de algún modo, independientemente de su rol dentro del programa *Coleccionando Sonrisas*.

Se considera un usuario a toda persona que participe en el proyecto y que pueda abrir una sesión a través de alguna de las herramientas que el sistema provee. Para

[1]El modelo Entidad-Relación nos permite diseñar bases de datos a partir de entidades y las relaciones que existen entre ellas [19]

una agente de este tipo se considerarán los siguientes atributos.

Nombre: Nombre de la persona.

Apellido: Apellidos (paterno y materno).

Fecha de nacimiento: Incluyendo el año, el mes y el día.

Teléfono: Número telefónico de contacto del donador.

Correo electrónico: Dirección de correo electrónico para contactar al donador.

País: Representa el país en donde vive el usuario.

Código postal: Será el código postal del domicilio del usuario.

Fecha de incorporación al programa: Año, mes y día en el que el donador se registra en el sistema.

Estado: Este atributo representa el estado de la cuenta del usuario. Los estados posibles son:

Inicial: Para usuarios en proceso de registro.

Suspendido: Para usuarios que no han realizado el último pago requerido.

Baja: Para cuentas inactivas.

Normal: Para cualquier otro caso.

Rol: Cada usuario deberá tener asignado uno de los roles que se describen en la Sección 2.1

Login: El nombre que identificará unívocamente al usuario dentro del sistema.

Contraseña: La contraseña que usará el usuario para abrir una sesión dentro del sistema.

3.1.2 Pirámide

Dado que más de un grupo de personas con intereses semejantes pueden participar en el programa *Coleccionando Sonrisas*, será necesario considerar la posibilidad de que exista más de una pirámide[2].

La información relevante de una pirámide será la que corresponda a los siguientes atributos:

Nombre: La palabra con la que se le identificará dentro del programa *Coleccionando Sonrisas*.

[2]La descripción de las pirámides se encuentra en la Sección 1.2.1

Altura: El número de niveles (estén completos o no) que se pueden apreciar dentro de la pirámide.

Amigos: El número de donadores registrados en la pirámide.

Saludo: El mensaje de bienvenida que verán los donadores registrados en esta pirámide al iniciar su sesión.

Último Corte: Fecha en que se llevó a cabo la última distribución de puntos para esta pirámide.

Sólo los usuarios con rol de donador podrán estar registrados en alguna pirámide.

3.1.3 Donador

Como se describe en la Sección 1.2.1, un donador es aquel que participa en el programa colaborando mensualmente con un monto específico de dinero a cambio del cual recibirá 10 puntos fijos además de los adicionales según el número y la distribución de los donadores que se registren después de él en la misma pirámide. Acerca de un donador, se considerará relevante la siguiente información:

Usuario: Todo donador debe estar asociado a un usuario.

Puntos Acumulados

Pirámide: Todo donador debe estar asociado a una pirámide.

Posición en la Pirámide: La posición se representará como un juego de coordenadas. La primera entrada hará referencia al nivel (comenzando por 1) y la segunda al número secuencial (comenzando por 0 y contando de izquierda a derecha).

3.1.4 Cuenta de débito

Dado que un donador podrá solicitar cargos automáticos a su cuenta de ahorro o tarjeta de débito será necesario preservar información acerca de las mismas. Para ello, será imprescindible contar con la siguiente información:

CLABE o Número: Se refiere a los 16 dígitos de la tarjeta de débito o a los 18 dígitos de la Clave Bancaria Estandarizada (CLABE) de la cuenta del donador.

Banco: Para poder generar los cargos a dicha cuenta, será necesario contar con el identificador del banco al que pertenece.

3.1.5 Tarjeta de Crédito

Por los mismos motivos que se describen en el apartado anterior, se deberán considerar indispensables los siguientes datos acerca de las tarjetas de crédito de los donadores:

Número: Los dieciseis dígitos del número de tarjeta.

Código: Los tres dígitos escritos en la parte trasera de la tarjeta, junto a la firma del titular.

Vigencia: La fecha de vencimiento de la tarjeta en el formato MM/AA.

3.1.6 Banco

Acerca de los bancos registrados en la base de datos de *Payworks®*, se tomarán en cuenta los siguientes datos:

Nombre : Nombre de la institución bancaria.

Identificador: Se asignará un número entero único definido por *Payworks®* a cada uno de los bancos registrados en el sistema que servirá para generar los archivos de cargo correspondientes.

3.1.7 Producto

En esta entidad se encapsularán los datos registrados acerca de los productos que la *FPN* mantiene en su almacén. Aunque la información relevante acerca de los productos se describe con detalle en la Sección 1.2.2, considerar esta entidad dentro del diseño de la aplicación facilitará que los datos recolectados a través del formulario correspondiente se registren en el sistema.

3.1.8 Alta

Esta entidad tiene como objetivo encapsular los datos obtenidos cuando un donante se registra utilizando su clave bancaria estandarizada o número de tarjeta de débito. Dado que el proceso de afiliación que debe llevarse a cabo necesita intervención de más de un usuario para poder completarse no se registrará inmediatamente

a los donadores en el árbol respectivo, en cambio, se creará un objeto alta que simbolizará el hecho e que un donador específico deberá ser registrado en un subárbol determinado en cuanto se complete su afiliación para pagos recurrentes. La información que contempla esta entidad es:

Usuario: Toda alta debe estar asociada al usuario cuyo registro está pendiente por aprobación bancaria.

Nombre del Árbol: Indicará el nombre árbol en donde se registrará al nuevo donador.

Nombre de referencia: Indicará el nombre de usuario de la raíz del subárbol donde se deberá registrar al nuevo donador, también es capturado por el mismo.

Estado: Indicará el estado en el que se encuentra un alta en el momento de la consulta.

El diseño de las altas contempla los siguientes estados posibles:

inicial: Cuando el alta se acaba de crear.

enviado: Cuando ya se ha generado el archivo de afiliación correspondiente.

preproceso: Cuando se ha completado la afiliación pero aún no se solicita pago por la misma.

proceso: Cuando ya se ha procesado la afiliación pero aún no se recibe pago por la misma.

rechazado: Cuando ya se ha procesado la afiliación pero fue rechazada por *Payworks®*.

pendiente: Cuando el cargo por afiliación ha sido rechazado por *Payworks®*.

Los objetos pertenecientes a esta entidad serán temporales, solamente permanecerán almacenados dentro del sistema mientras se completa el proceso descrito en la Sección 2.2.1. Al finalizar cada proceso de afiliación se eliminará permanentemente el alta.

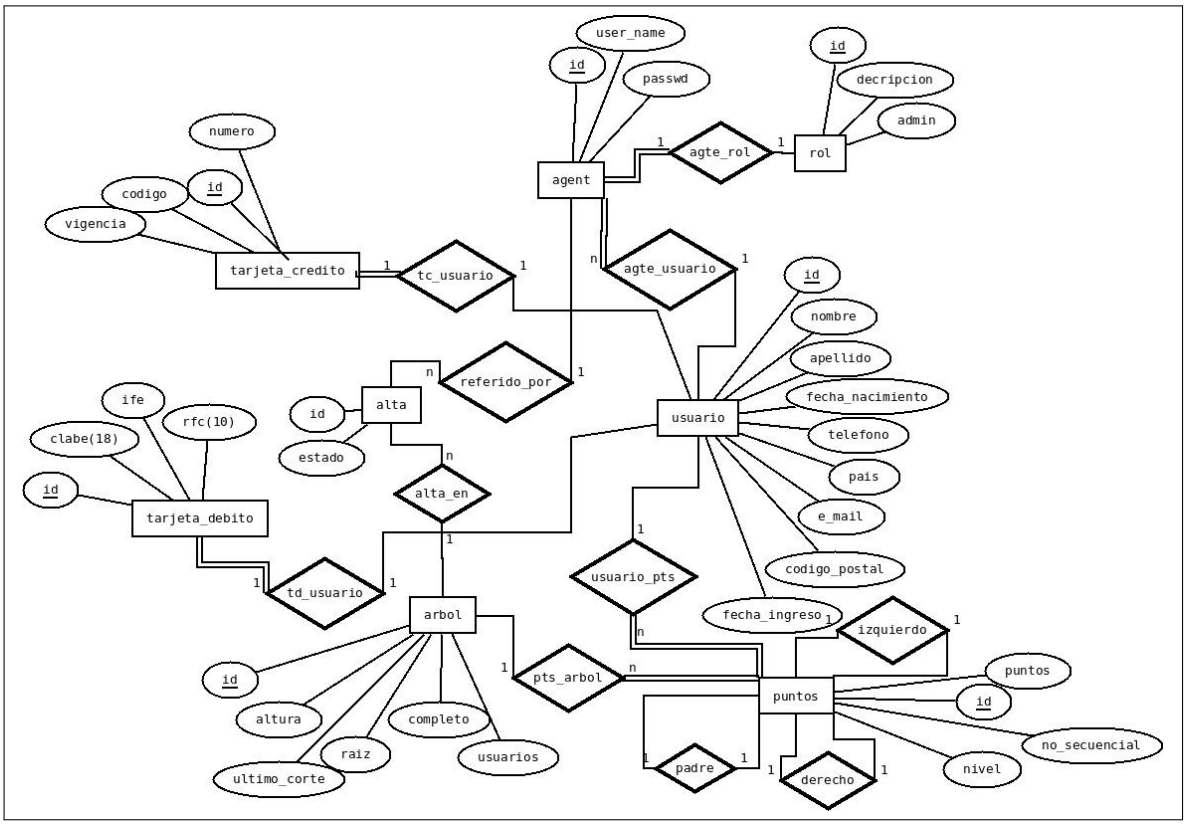


Figura 3.1: Diagrama Entidad-Relación del sistema.

3.2 Diseño de base de datos

Procederemos ahora a describir el diseño de la base de datos de la aplicación. Para explicar mejor el diagrama Entidad-Relación de la Figura 3.1, abordaremos esta descripción en un enfoque tabular, al describir cada Tabla[3] se especificarán tanto los atributos (columnas) y tipos de dato almacenados como el agente o agentes involucrados en las mismas. Además analizaremos los supuestos correspondientes sobre la información que contienen.

Para facilitar su lectura y comprensión describiremos a una relación listando sus atributos. Para cada uno de ellos, anotaremos el nombre, el tipo de dato y la

[3]En el resto de este capítulo usaremos Tabla (con esta tipografía) para referirnos a una relación dentro de la base de datos, de aparecer esta palabra sin mayúscula inicial y en la tipografía del resto del texto no deberá asociarse con la base de datos.

Nombre	***Tipo***	***Descripción***
id	numérico	Identificador unívoco del usuario en la base de datos.
descripcion	texto	

Cuadro 3.1: Atributos de la Tabla `rol`

correspondiente relación que guarda con la información especificada en la Sección 3.1.

3.2.1 Tabla `rol`

Ésta será un catálogo en el que se listan los roles que puede tener un usuario dentro del sistema según lo descrito en la Sección 2.1. Cuenta solamente con dos campos, a saber, el identificador numérico del rol y la descripción del mismo. Los valores posibles son:

1: Donador.
2: Operador técnico.
3: Supervisor de datos sensibles.

El Cuadro 3.1 muestra los atributos que definen a cada `rol`.

3.2.2 Tabla `usuario`

Ésta representará al agente homónimo, cada uno de los registros en ella se interpretará como una persona registrada dentro del programa *Coleccionando Sonrisas*. En el Cuadro 3.2 se encuentra la descripción de los atributos.

Cabe destacar que en esta Tabla se incluye una referencia hacia otra, la presencia de este apuntador queda justificada en el tipo de relación que guardan `usuario` y `estado_usuario` que es varios a uno con participación total de la primera y parcial de la segunda.

3.2.3 Tabla `agent`

Se considera un agente como la relación que existe entre un usuario y sus correspondientes nombre de usuario (`login`), `contraseña` y `rol`. Este diseño permite que una misma persona pueda asumir varios roles dentro del sistema dependiendo de qué datos específicos use para iniciar la sesión.

Nombre	***Tipo***	***Descripción***
id	numérico	Identificador unívoco del usuario en la base de datos.
nombre	texto	
apellido	texto	
fecha_nacimiento	fecha	Fecha de nacimiento del usuario.
telefono	numérico	
email	texto	
pais	texto	
codigo_postal	numérico	
fecha_ingreso	fecha	
estado_id	numérico	Referencia al estado del usuario.

Cuadro 3.2: Atributos de la Tabla `usuario`

Nombre	***Tipo***	***Descripción***
id	numérico	Identificador unívoco del agente en la base de datos.
usuario_id	numérico	Referencia al usuario asociado a este agente.
login	texto	Identificador alfanumérico unívoco del usuario.
passwd	texto	Contraseña elegida por el usuario para iniciar sesión.
rol_id	numérico	Referencia al rol en el sistema al que corresponde este agente.

Cuadro 3.3: Atributos de la Tabla `agent`

Esta Tabla guarda una relación uno varios a varios con `usuario`, esto es, un `usuario` puede estar relacionado con diversos agentes y viceversa. Por otro lado, existe una asociación varios a uno con `rol` pues es posible que existan muchos agentes con el mismo rol mientras que no puede existir agente con más de un rol.

El Cuadro 3.3 muestra los atributos y estructura de `agente`.

3.2.4 Tabla `estado_usuario`

Contando solamente con dos atributos sirve para representar el estado en el que se encuentra la cuenta de un usuario con base en las siguientes reglas:

Inicial: El registro del usuario está en proceso de afiliación bancaria.

Suspendido: La cuenta estaba activa pero no se ha registrado el último pago.

Nombre	*Tipo*	*Descripción*
id	numérico	Identificador unívoco del usuario en la base de datos.
nombre	texto	

Cuadro 3.4: Atributos de la Tabla `estado_usuario`

Baja: La cuenta está cancelada.

Normal: La cuenta está activa y sin aportaciones pendientes.

Los atributos de esta Tabla se pueden apreciar en el Cuadro 3.4

3.2.5 Tabla `puntos`

Aquí se representa a la entidad donador. Se trata de una Tabla con referencias hacia sí misma pues, al mismo tiempo, se representan aquí los nodos de los árboles binarios correspondientes a las entidades pirámide descritas previamente.

Además de conservar referencias hacia el padre y los hijos, para facilitar el acceso a los datos se agregan doss atributos especiales, a saber, el nivel y el número secuencial. Estos dos números representan las coordenadas del nodo dentro del árbol asumiendo que está completo. De modo que la raíz tiene nivel 0 y número secuencial 0, en los hijos izquierdos se cumple $ns = 2^{np} * nsp$ y en los hijos derechos se cumple $ns = 2^{np} * nsp + 1$; donde ns es el número secuencial, np es el nivel del padre y nsp es el número secuencial del padre.

Como veremos más adelante, la incorporación de estos atributos permite implementar un algoritmo iterativo además de eficiente en tiempo y espacio para calcular la cantidad de puntos que se debe agregar a cada nodo en el árbol.

En términos de las referencias a registros de las Tablas `usuario` y `arbol` se debe mencionar que `puntos` participa en una relación uno a uno con `usuario`, sin embargo, la participación de la primera es total mientras que la de la segunda es parcial. Dado que ocurre lo mismo con `arbol`, se decidió diseñar esta Tabla con referencias hacia las otras.

En el Cuadro 3.5 se describen con detalle los atributos y estructura de esta Tabla.

Nombre	*Tipo*	*Descripción*
id	numérico	Identificador unívoco del nodo en la base de datos.
arbol_id	numérico	Referencia al árbol al que pertenece este nodo.
nivel	numérico	Distancia (en niveles) hasta la raíz del árbol.
no_secuencial	numérico	Posición horizontal dentro del árbol.
padre	numérico	Referencia al nodo padre (es nula para la raíz).
izquierdo	numérico	Referencia al hijo izquierdo (puede ser nula).
derecho	numérico	Referencia al hijo derecho (puede ser nula).
puntos	numérico	Total de puntos acumulados por el usuario. asociado a este nodo
usuario_id	numérico	Referencia al usuario asociado a este nodo
activo	lógico (verdadero o falso)	Referencia al nodo padre (es nula para la raíz)

Cuadro 3.5: Atributos de la Tabla `puntos`

3.2.6 Tabla `arbol`

Mientras que la mayor parte de la estructura de las pirámides descritas en la Sección 3.1.2 queda implementada en la Tabla `puntos` y dado que el sistema contempla la participación de usuarios en diversas pirámides según su procedencia o intereses en común, se consideró natural separar la información acerca de ellas en una Tabla distinta.

Las pirámides tienen un número máximo de niveles, este número lo establece la FPN al momento de crearlas dependiendo del límite de puntos adicionales que el donador en la raíz de la misma pueda recibir, suponiendo que ha llegado a construir un árbol completo.

En el Cuadro 3.6 se muestra la estructura de los árboles dentro de la base de datos.

3.2.7 Tabla `tarjeta_debito`

En el Cuadro 3.7 se muestra la estructura con la que se representarán el la base de datos las cuentas de débito de los donadores. Como se describió en la Sección 3.1.4

Nombre	***Tipo***	***Descripción***
id	numérico	Identificador unívoco del árbol en la base de datos.
altura	numérico	Distancia al nodo más profundo del árbol.
completo	lógico	Es verdadero si el árbol alcanzó ya su máximo número de niveles
raiz	numérico	Identificador del nodo que representa la raíz de este árbol.
altura_maxima	numérico	Número máximo de niveles que puede alcanzar este árbol.
ultimo_corte	fecha	Fecha en el que se llevó a cabo la última distribución mensual de puntos sobre este árbol.
nombre	texto	Nombre de la pirámide a la que representa este árbol según la FPN.
imagen	texto	Ruta relativa (dentro del servidor) de la imagen que se desplegará en el perfil de los donadores registrados en este árbol.
saludo	texto	Mensaje que se desplegará en el perfil de los donadores registrados en este árbol.

Cuadro 3.6: Atributos de la Tabla `arbol`

Nombre	*Tipo*	*Descripción*
id	numérico	Identificador unívoco de la tarjeta en la base de datos.
usuario_id	numérico	Referencia al usuario propietario de esta tarjeta.
clabe	text	18 números de la CLABE de la cuenta o los 16 del plástico.
banco_id	numérico	Referencia al banco que brinda la tarjeta o cuenta.

Cuadro 3.7: Atributos de la Tabla `tarjeta_debito`

los registros en esta Tabla deberán contener los detalles de la cuenta o tarjeta de débito de los donadores.

Las Tablas `banco` y `tarjeta_debito` participan en una relación varios a uno en la que la participación de la tarjeta es total. Motivo por el cuál se almacena la referencia al banco dentro de la tarjeta. Por otro lado, la relación con la Tabla `usuario` es uno a uno pero mientras que la participación de `tarjeta_debito` es total, la del `usuario` es parcial, lo que justifica también la presencia de la referencia al usuario en la tarjeta.

3.2.8 Tabla `tarjeta_credito`

Ilustrados en el Cuadro 3.8 se describen los atributos de la entidad definida en la Sección 3.1.5. Como se puede apreciar, los números de tarjeta y de seguridad de la misma no se representan con texto ni con sus respectivos valores numéricos. Esta representación se diseñó asumiendo que al momento de almacenar dichos números en la base de datos ya están encriptados y los datos almacenados son simples secuencias de bytes. El mecanismo usado para lograr este objetivo se describirá con detalle en la Sección 4.1.

3.2.9 Tabla `banco`

Tal y como se ha descrito en la Sección 3.1.6, sólo se almacenarán el nombre y el código de identificación de cada banco en la base de datos del programa *Coleccionando Sonrisas*, en el Cuadro 3.9 se muestra la estructura con la que almacenará la información acerca de estas entidades. Aunque dentro de su estructura no es posible apreciar referencias hacia otras Tablas es importante destacar que `banco` está

Nombre	*Tipo*	*Descripción*
id	numérico	Identificador unívoco de la tarjeta en la base de datos.
numero	secuencia de bytes	Los 16 dígitos codificados
codigo_seguridad	secuencia de bytes	Los 3 dígitos de la parte trasera de la tarjeta codificados.
vigencia	texto	Fecha de vencimiento representada en el formato AA/MM.
tipo_tarjeta	Texto	N para nacional y E para extranjera.
ultimo_impacto	Fecha	Fecha y hora en la que se hizo el último cargo a la tarjeta.

Cuadro 3.8: Atributos de la Tabla `tarjeta_credito`

Nombre	*Tipo*	*Descripción*
id	numérico	Identificador unívoco del banco en la base de datos.
nombre	texto	
codigo	numérico	Identificador numérico asignado al banco por Banorte.

Cuadro 3.9: Atributos de la Tabla `banco`

relacionada con `tarjeta_debito`, esta relación es de uno a varios con participación parcial de la primera y total de la segunda.

3.2.10 Tabla `alta`

Como versa en la Sección 3.1.8 se almacenará cierta información temporalmente mientras se completa la afiliación bancaria de un nuevo donador. El esquema descrito en el Cuadro 3.10 muestra con detalle la estructura que guardarán los objetos de esta entidad en la base de datos.

3.2.11 Tabla `producto`

Retomando la definición de la entidad vertida en la Sección 3.1.7, se muestra en el Cuadro 3.11 el detalle de cada uno de los atributos que esta entidad conservará

Nombre	***Tipo***	***Descripción***
id	numérico	Identificador unívoco del alta en la base de datos.
usuario_id	numérico	referencia al usuario que deberá quedar registrado al finalizar la afiliación.
padre	texto	Nombre del usuario de referencia.
arbol	texto	Nombre del árbol deseado.
estado	texto	Fase en la que se encuentra la afiliación.

Cuadro 3.10: Atributos de la Tabla `alta`

dentro de la base de datos.

Cabe señalar que el atributo `clasifica` se diseñó originalmente como un atributo multivaluado, el significado de cada uno de los valores que puede tener este atributo es interpretado por la aplicación de acuerdo a la siguiente lista:

1.- Entretenimiento
2.- Deportes
3.- Tecnología
4.- Hogar
5.- Cultura
6.- Sonrisas
7.- Vivenciales

Se presenta una situación similar con el atributo `rating`, cuyos valores se interpretan de acuerdo a la siguiente lista:

0.2 Muy Baja
0.4 Baja
0.6 Promedio
0.8 Alta
1.0 Muy Alta

Nombre	***Tipo***	***Descripción***
id	numérico	Identificador unívoco del banco en la base de datos.
nombre	texto	
clasifica	numérico	Categoría a la que pertenece el producto
descripcion	texto	
existencia	numérico	Número de objetos de este producto que están disponibles para su intercambio
vmercado	numérico	Valor comercial del producto en pesos
rating	numérico	Demanda esperada
imagen	texto	Ruta relativa de la imagen del producto
ingreso	fecha	Fecha en la que se registró el producto

Cuadro 3.11: Atributos de la Tabla `producto`

Capítulo 4

Diseño del Sistema

Tomando en cuenta que el presente texto es un reporte de trabajo profesional, en este capítulo analizaremos solamente los aspectos del diseño en los que se aplicaron conocimientos especializados que se adquieren en la Licenciatura en Ciencias de la Computación. El diseño global del sistema queda fuera del alcance de este reporte pues no fue labor exclusiva del sustentante.

4.1 Seguridad en nivel de datos

La *FPN* considera que los números de tarjeta de crédito así como los códigos de seguridad de las mismas son información privilegiada a la que sólo los operadores técnicos, bajo la estricta supervisión de un representante del patronato de la Fundación, podrían tener acceso.

Para garantizar que, a pesar de ser capaces de manipular la base de datos, los usuarios administradores no tuvieran acceso a dicha información se decidió almacenarla de manera encriptada. En esta sección describiremos el algoritmo de encripción que fue utilizado para sobre estos datos, una adaptación del algoritmo de compresión de Huffman creada por el sustentante que permite, además de comprimir, proteger la información con una contraseña.

4.1.1 Algoritmo de compresión de Huffman

Dado que se trata de información que puede entrar al sistema (a través de los formularios de registro de usuarios) o salir del mismo (a través de la interfaz de la que disponen los operadores técnicos para generar un archivo de cobranza a tarjetas de crédito) fue necesario diseñar un esquema de encripción que hiciera uso de un código unívocamente decodificable, esto es, que permitiera al programa encriptar la información pero también desencriptarla.

El algoritmo de Compresión de Huffman[6] provee un método para codificar y decodificar que consta de un código unívocamente decodificable.

La codificación de Huffman consiste en generar un árbol binario con base en una tabla de frecuencias de aparición de los caracteres dentro del mensaje a encriptar, en cada nodo se almacena una cadena de caracteres y la suma de las frecuencias de los caracteres en ella según la tabla. De esta manera, en las hojas del árbol permanecen los caracteres y sus frecuencias.

A estos árboles se les conoce como **Árboles de Huffman** y permiten codificar un mensaje carácter por carácter.

Para codificar un mensaje se debe leer el mismo carácter por carácter y buscarlo dentro del árbol escribiendo un bit cero por cada desplazamiento a la izquierda y un bit uno por cada desplazamiento a la derecha.

Por otro lado, el proceso de decodificación lee el mensaje previamente codificado bit por bit desplazándose por el árbol binario según las reglas descritas previamente, al llegar a una hoja, se escribe el carácter almacenado en ella y la siguiente búsqueda comienza desde la raíz llegando así al original después de haber leído todos los bits del mensaje codificado.

4.1.2 Huffman con contraseña

Siendo necesario que el sistema impidiera que un operador técnico desencriptara los números de tarjeta de crédito almacenados en la base de datos, se implementó un esquema de seguridad en el se establece una contraseña de exactamente 10 dígitos hexadecimales, ésta es indispensable para generar el archivo de cobranza a tarjetas de crédito y está programada dentro del código del sistema.

Dígito	***Frecuencia***
0	14
1	10
2	11
3	12
4	13
5	1
6	2
7	3
8	4
9	5

Cuadro 4.1: Tabla de frecuencias para la clave EABCD12345

Dicha clave alfanumérica se usa como llave para encriptar y desencriptar la información sensible y sólo la poseen los representantes del patronato de la Fundación. De este modo, cuando un operador técnico necesita generar el archivo de cobranza mensual es necesario que esté presente algún miembro del patronato.

Internamente, se utiliza dicha secuencia de números para representar las frecuencias de cada uno de los 10 dígitos decimales, por ejemplo, una contraseña como EABCD12345 representaría la tabla de frecuencias que se puede apreciar en el Cuadro 4.1.

A partir de la tabla de frecuencias derivada de la contraseña se construye un árbol de Huffman con el que se llevan a cabo los procesos de codificación y de-codificación de datos. Dado que la única manera en que se podrían generar árboles distintos para la misma contraseña implicaría que hay varios dígitos con la misma frecuencia, los empates se resuelven según el valor numérico de los dígitos, por ejemplo si el dígito **A** y el dígito **9** tienen la misma frecuencia, se considerará que **9** va antes que **A** (como si la frecuencia de **9** fuera menor que la de **A**).

De esta manera, aún cuando se cuente con acceso a la base de datos del sistema y se tenga experiencia y destreza en el manejo y administración de bases de datos relacionales, los usuarios administradores sólo podrán obtener secuencias pequeñas de bits al tratar de extraer la información privilegiada.

4.2 Algoritmo de distribución de puntos

El objetivo principal del programa *Coleccionando Sonrisas* es conseguir el mayor número posible de donadores para la *FPN*, en estos términos se estimó que en un futuro no muy lejano el sistema trabajaría sobre pirámides de 10 a 16 niveles atendiendo a más de 32,000 usuarios.

Tomando en cuenta el panorama descrito anteriormente aunado al hecho de que se dispone de una ventana de tiempo muy estrecha para ejecutar la distribución mensual de puntos, fue necesario diseñar un algoritmo para que el sistema llevara a cabo esta tarea en poco tiempo. En esta sección se aborda el diseño de este algoritmo.

4.2.1 Proceso de distribución de puntos

Como se describió en la Sección 1.2.1 mensualmente los donadores reciben una cantidad adicional de 10 puntos por cada 2^n (con $n >= 1$) donadores registrados en el nivel `n` de su subárbol, esto es, por cada nivel completo.

Poniéndolo en términos muy simples podríamos describir el proceso genérico de distribución de puntos sobre un árbol de la siguiente manera:

1. Para cada nodo en el árbol:
   1.1 Calcular el número k de niveles completos en el subárbol del cuál es raíz.
   1.2 Abonar $10 + 10 * k$ puntos a la cuenta del usuario correspondiente.

Lo que reduce el problema a calcular en número de niveles completos en un árbol.

La prospectiva elaborada por la *FPN* indica una tendencia marcada hacia el balance de las pirámides. A partir de esto, es válido suponer que los árboles binarios que las representan estarán balanceados en el caso promedio.

En dicho escenario, el proceso descrito anteriormente tiene un costo elevado en tiempo pues sólo se puede calcular el número de niveles completos en un árbol visitando todos sus nodos. Como es necesario hacer esto para cada nodo en el árbol la complejidad de este acercamiento ingenuo sería de $O(n^2)$.

4.2.2 Algoritmo de búsqueda en amplitud con Propagadores

En esta sección se expone el algoritmo cuya complejidad en el caso promedio, según la prospectiva de la *FPN*, es $O(n * log_2(n))$. Fue diseñado por el sustentante haciendo uso de los conocimientos adquiridos en la licenciatura en Ciencias de la Computación y supone una notable mejora a la estrategia ingenua descrita en la sección precedente.

El algoritmo está basado en la estrategia de búsqueda en amplitud o *Breadth First Search, BFS*[4], esto es, realiza un recorrido del árbol binario por niveles. Así, si comenzamos desde el nivel más profundo, podemos aprovechar el recorrido de cada nivel para calcular los puntos adicionales de cada nodo en los niveles superiores.

Para el diseño de este algoritmo, se consideraron los siguientes supuestos:

- Cada nodo tiene una referencia a su padre.
- Cada nodo tiene una referencia a cada uno de sus hijos.
- Existe una operación que en tiempo constante obtiene al nodo en las coordenadas (i, j) del árbol.
- Existe una operación que permite abonar 10 puntos a un nodo activo dentro del árbol[1]

Propagador

Llamaremos propagador a una estructura de datos que nos permitirá abonar puntos adicionales a todos los ancestros de un conjunto determinado de nodos del mismo nivel.

Podríamos pensar en los propagadores como ”mecanismos de retraso” que funcionarán de manera similar a los que hay en las manecillas de un reloj, esto es, la manecilla de los minutos no avanza hasta que la manecilla de los segundos da una vuelta completa y lo mismo sucede con la manecilla de las horas respecto a la de los minutos[2]. Así, al hablar de propagadores es necesario establecer tres cosas primordialmente:

[1] Sólo se abonan puntos si el usuario asociado con el nodo se encuentra en estado inicial o normal.

[2] Aunque, para que esta analogía fuera más precisa, la relación estaría entre la manecilla de las horas y la de los segundos, la manecilla de las horas avanza cada que la de los segundos da 3600 vueltas.

1. El dato pasivo.
2. El dato activo.
3. El intervalo de retraso.

En donde el intervalo de retraso nos dice cuántos cambios del dato activo deben ocurrir para que se presente uno en el dato pasivo.

Dejando atrás las analogías, un propagador consta de dos referencias: una al ancestro a quien se le abonarán puntos adicionales (dato pasivo) y otra al sucesor actual (dato activo). De esta manera, si n_a es el nivel del ancestro actual y n_s el del sucesor actual[3], el intervalo de retraso del propagador será de $2^{(n_s - n_a)}$, esto es, el propagador "cambiará de ancestro" cada $2^{(n_s - n_a)}$ sucesores. Dado que el ancestro actual es la raíz de un subárbol, si éste tiene $2^{(n_s - n_a)}$ nodos en el nivel n_s entonces podemos inferir que dicho ancestro tiene "completo" el nivel n_s por lo que merece 10 puntos adicionales.

En adelante identificaremos a los propagadores usando sus dos componentes principales. Así, si hablamos de un propagador en el que el dato pasivo son los nodos del nivel k y el dato activo son los nodos del nivel n diremos que se trata de un propagador de k sobre n o P_n^k.

El diseño de un propagador contempla los siguientes datos:

- Nivel asociado al propagador, k.
- Nivel de los nodos sobre los que se desplaza el propagador, n.
- Meta o intervalo de retraso, 2^{n-k}.
- Contador de posición dentro del intervalo de retraso del propagador, $[0, 2^{n-k}$.
- Número secuencial del ancestro actual.
- Marcador de aplicación (para saber si el ancestro actual recibirá o no puntos adicionales).

Tomando en cuenta este diseño, con un propagador del 2 sobre 4 podría hacer un recorrido sobre los 16 nodos del nivel 4 de uno en uno, cambiando de ancestro (del nivel 2) cada 4 sucesores, en la Figura 4.1 se muestra una representación gráfica del recorrido factible usando dicho propagador, se marcan los ciclos con colores distintos en la secuencia gris, verde, azul y amarillo.

[3] Al hablar de sucesores y ancestros en un árbol binario garantizamos que $n_s > n_a$

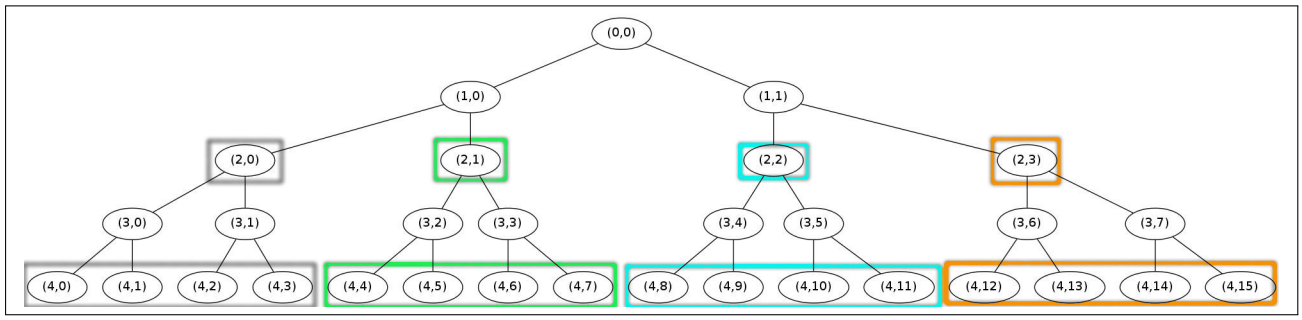


Figura 4.1: Recorrido de un propagador de nivel 2 sobre nodos del nivel 4

Recorrido de un nivel

Gracias a los propagadores es posible implementar un algoritmo tal que, al recorrer un nivel determinado, abone los puntos adicionales que los nodos de dicho nivel brindan a sus respectivos ancestros, este algoritmo supone la existencia de una lista de propagadores, uno por cada nivel anterior al actual. Si le llamamos $distr(k, a)$, donde k es el nivel a recorrer y a es el árbol donde se hará el recorrido lo podemos caracterizar con el pseudo-código mostrado en el Algoritmo 1.

4.2.3 Recorrido sobre un árbol

Mezclando lo expuesto anteriormente, el Algoritmo 2 llevaría a cabo el recorrido de un árbol distribuyendo todos los puntos adicionales que cada nodo debe recibir.

Para hacer el recorrido de un árbol balanceado a de altura k se hacen exactamente k llamadas a $distr(k_i, a)$, en cada una se disminuye en uno el nivel. El ciclo más amplio que hace el Algoritmo 1 es de 2^{k_i} pasos donde k_i es el nivel recorrido, en otras palabras, en cada llamada se recorre el número total de nodos que tendría a en su nivel k_i si estuviera completo. El número de iteraciones se podría expresar como $n' = n + m = 2^k - 1$ siendo $m > 0$ si el árbol no está completo y n el número total de nodos en a. Ahora bien, en cada iteración se recorre una lista de propagadores de tamaño k_i, el nivel actual. Considerando que k_i recorre los valores $[k-1, k-2, .., 0]$, pero el recorrido para el nivel cero se hace en tiempo constante, podemos expresar el número total de iteraciones que hace el algoritmo con la expresión $\left(\sum_{i=1}^{k-1} (k-i)2^{k-i}\right) + 1$ que

input : Nivel k sobre el que se propagarán los puntos.
input : Árbol a que se va a recorrer.
output: Al terminar, se habrán distribuido los puntos adicionales sobre todos los nodos en el nivel k del árbol a.

```
begin
    for (i := 0 → k − 1) do
        P_k^i := nuevo propagador de i sobre k;
        P_k^i.secAncestro := 0;
        P_k^i.meta := 2^{k−i};
        P_k^i.aplica := true;
        P_k^i.posicion := 0;
    end
    for (j := 0 → 2^k) do
        nodoActual := buscar(j, k);
        for (i := 0 → k − 1) do
            P_k^i.aplica := P_k^i.aplica && nodoActual! = null;
            P_k^i.posicion := P_k^i.posicion + 1;
            if (P_k^i.meta == j) then
                if (P_k^i.aplica) then
                    abonaPuntos(buscar(0, P_k^i.secAncestro));
                end
                P_k^i.posicion := 0;
                P_k^i.aplica := true;
                P_k^i.secAncestro := P_k^i.secAncestro + 1;
            end
        end
    end
end
```

Algoritmo 1: Distribución de puntos sobre un nivel con propagadores.

podemos acotar por arriba de la siguiente manera

$$\left(\sum_{i=1}^{k-1}(k-i)2^{k-i}\right)+1 <= \left(k\sum_{i=1}^{k-1}2^{k-i}\right)+1 = \left(k\sum_{i=1}^{k-1}2^{i}\right)+1 <= k\sum_{i=0}^{k-1}2^{i}$$

como $k = log_2(n')$[4] y $\sum_{i=0}^{k-1}2^i = 2^k - 1$ la complejidad del Algoritmo 2 es

$$O\left(n' * log_2(n')\right)$$

.

input : Árbol A sobre el que se propagarán los puntos
input : k altura del árbol A
output: Al terminar, el se habrán distribuido todos los puntos adicionales a todos los nodos del árbol A

```
begin
    for i := k →0 do
        distr(i, A);
    end
end
```

Algoritmo 2: Distribución de puntos sobre un árbol.

4.3 Arquitectura de herramientas y servicios

Varios equipos de trabajo integrados por diseñadores web, diseñadores gráficos y programadores estuvieron involucrados en el desarrollo del proyecto, sin embargo, la falta de un arquitecto de bases de datos hizo necesaria la participación de un profesional cuya formación incluyera conocimientos relacionados con diseño e implementación de herramientas de persistencia y acceso flexible y confiable a los datos.

Dada la heterogeneidad de las metodologías y las estrategias a las que cada equipo estaba acostumbrado se decidió adoptar una arquitectura de tres capas para

[4]Pues la altura de un árbol binario completo es $log_2(n)$ donde n es el número total de nodos del mismo.

el diseño de la aplicación, a saber: datos[5], acceso a datos y vista/control[6]. Siendo una simplificación del patrón de diseño Modelo/Vista/Controlador (MVC), adoptar esta estrategia permitió a los responsables de cada capa completar sus tareas sin adquirir capacitación adicional.

Por otro lado se diseñaron aplicaciones independientes para los procesos asistidos descritos en la Sección 2.4 así como para la generación de archivos de cobranza a tarjetas de crédito cuyo uso requiere la presencia de un representante del patronato de la *FPN*.

4.3.1 Diseño de Clases

Todas las herramientas diseñadas que se describen en las secciones subsecuentes están construidas sobre una plataforma que sigue el patrón de la Programación Orientada a Objetos. Se describirán en esta sección cada una de las clases sobre las que se apoya el resto del sistema. Describiremos a grandes rasgos cada una de ellas, los detalles sobre los métodos y atributos se vierten en el diagrama de clases de la Figura 4.2.

Arbol

Esta clase permite encapsula la estructura de las pirámides, implementa los métodos necesarios para llevar a cabo la distribución de puntos sobre sus nodos.

Nodo

Aquí se abstrae la entidad homónima permitiendo al sistema actualizar los datos de cada nodo durante el recorrido de distribución de puntos.

[5]La capa de datos se describe en el Capítulo 3

[6]Dentro de esta capa se encuentra el software que gestiona las peticiones de los clientes (navegadores) y, a partir de las herramientas que provee la capa de acceso a datos, genera las respuestas necesarias en forma de páginas web.

Usuario

En esta clase se encapsulan los datos personales que todo donante captura al registrarse en el sistema y que pueden ser desplegados en alguna de las pantallas del sistema.

GeneradorArchivos

El diseño de esta clase pretende concentrar las funciones que generan archivos de intercambio de datos con *Payworks®*. Provee métodos para generar diversos archivos de programa con base en diversos parámetros.

Cola

Representa a la estructura de datos *First-In-First-Out*, FIFO, que se usará para la implementación del algoritmo de distribución de puntos.

Notificador

Abstracción de las herramientas de envío de correos electrónicos basada en plantillas HTML.

DBManager

Implementación completa de la capa de acceso a datos. En esta clase se concentran todas las herramientas que permiten intercambio de información entre dicha capa y las superiores.

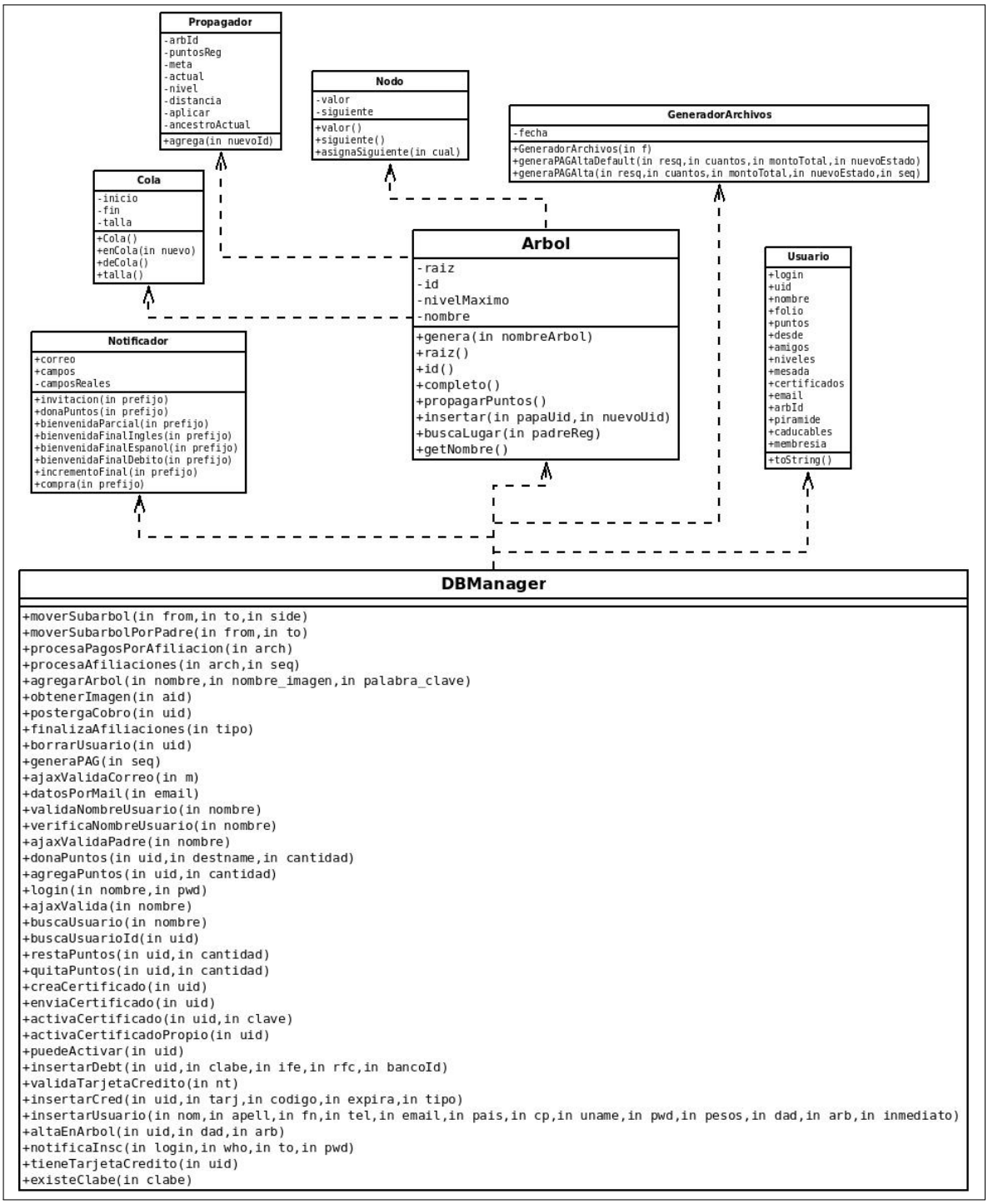


Figura 4.2: Diagrama de clases del sistema

4.3.2 Acceso a datos desde la aplicación

Considerando el hecho de que la mayoría de los programadores involucrados en el proyecto no estaban familiarizados con la Programación Orientada a Objetos (POO) el diseño de la capa de acceso a los datos se redujo a una sola clase en la que se pueden encontrar todos los métodos que se requirieron para completar el desarrollo del proyecto. Dicha clase contiene toda la funcionalidad necesaria para consultar, modificar y crear las entidades almacenadas en la base de datos. A continuación describiremos cada una de las operaciones diseñadas para esta capa.

Inserción de tarjetas bancarias

Desde la perspectiva del acceso a datos a través de la aplicación, la inserción de tarjetas bancarias es un proceso adjunto a la de usuarios, esto es, sólo se crearán registros en `tarjeta_debito` o `tarjeta_credito` al momento de crear los correspondientes en `usuario`.

Mientras que para el caso de las cuentas o tarjetas de débito la lógica consiste únicamente en crear registros en la Tabla correspondiente, es necesario profundizar sobre la inserción de tarjetas crédito pues justo en esta herramienta se debe aplicar el algoritmo de encripción descrito anteriormente.

Se describe a continuación el proceso de registro de tarjetas de crédito con el Algoritmo 3 asumiendo la existencia de una las siguientes funciones y procedimientos:

***encripta*(n):** que recibe como parámetro una secuencia de números decimales n y regresa la secuencia de bits resultante de aplicar el algoritmo de Huffman con contraseña utilizando como llave aquella elegida por el representante del patronato de la *FPN*.

***insertar*:** que toma los parámetros descritos en el Cuadro 3.8, excepto el identificador, y crea el registro correspondiente en la base de datos.

***now*():** que genera la fecha y hora actuales.

Inserción de usuarios

La herramienta de inserción de usuarios fue diseñada considerando su utilidad en el formulario de registro de donadores. Asumiendo que dicho formulario cumple

input : n número de tarjeta de crédito
input : k código CVV de la tarjeta de crédito
input : v fecha de vigencia de la tarjeta de crédito
input : t tipo de tarjeta (nacional o extranjera)

begin
    $ultimoImpacto := now()$;
    $nTCodificado := encripta(n)$;
    $cvvCodificado := encripta(k)$;
    $insertar(nTCodificado, cvvCodificado, v, t, ultimoImpacto)$;
end

Algoritmo 3: Caracterización del algoritmo para insertar tarjetas de crédito en la base de datos.

con las especificaciones descritas en la Sección 2.2.1, se contemplan los siguientes parámetros:

- Nombre
- Apellido
- Fecha de nacimiento
- Teléfono
- Correo Electrónico
- País de Origen
- Código Postal
- Nombre de usuario
- Contraseña de acceso
- Moneda(pesos/dólares)
- Usuario de referencia
- Pirámide de referencia
- Indicador de inmediatez

La finalidad de esta herramienta es crear las entidades usuario y donador dentro del sistema. En consecuencia, deberá agregar los registros necesarios a la base de datos sobre las Tablas `usuario` y `agente`. Dependiendo del indicador de inmediatez se crearía un registro en `alta`. Esto es, si dicho marcador es verdadero se entenderá que el registro del donante debe completarse creando el registros correspondiente en `puntos`, en otro caso, sólo se creará un registro en `alta` en estado *inicial* que posteriormente, cuando se complete la afiliación bancaria, se convertirá en un registro en `puntos`.

Vinculación de donadores a pirámides

Como se menciona en la sección anterior, después de la inserción de un usuario se debe llevar a cabo en algún momento la asociación de su cuenta con alguna de las pirámides registradas en el sistema. Aunque se trata de una herramienta auxiliar, es menester observar los detalles de su diseño. Se considera que tomará como entrada dos datos, a saber, el identificador del nuevo usuario y el nombre de usuario que éste puso como referencia al registrarse. Cabe destacar que el segundo de los parámetros puede representar también el nombre de la pirámide a la que el donante desea ser incorporado. La herramienta buscará primero la cadena proporcionada en el catálogo de nombres de usuario, de existir algún usuario como resultado, el nuevo donante quedará registrado en el primer lugar disponible[7] dentro del subárbol del que el usuario hallado es raíz. Si no existe un usuario con el nombre de usuario proporcionado será necesario buscar la cadena dentro del catálogo de nombres de pirámides, de existir una pirámide con dicho nombre, se incorporará al nuevo donador en el primer lugar disponible de la misma. En el caso en que no existan usuarios ni pirámides asociados a la cadena recibida se registrará al nuevo donante en el primer lugar disponible de la pirámide "general".

Búsqueda de usuarios

El diseño de esta capa contempla búsquedas de usuarios parametrizadas por los atributos `login` e `id`. En ambos casos la herramienta debe regresar un objeto que encapsule todos los atributos del usuario encontrado.

Autenticación de usuarios

Consistirá en una función que recibiendo como parámetro el nombre de usuario y contraseña de un donante devuelva alguno de los siguientes valores:

Número 0 Si el nombre de usuario no existe o la contraseña no coincide con la almacenada en la base de datos.

Número 2 Si los datos de acceso son correctos pero existe una alta asociada al usuario en cuestión.

[7] Este lugar se busca de arriba hacia abajo y de izquierda a derecha usando el algoritmo BFS

Objeto Usuario Si los datos de acceso son correctos.

Validación de nombre de usuario

Procurando el mejor funcionamiento de la aplicación, se considera como restricción sobre los nombres de usuario que sólo puedan comenzar con letras y no contengan caracteres como espacios en blanco, signos de puntuación, símbolos como #, y & ni caracteres acentuados. Para garantizar que un donador no pueda registrar un login inválido se diseño una herramienta que recibe una cadena como parámetro regresando verdadero si ésta cumple con la restricción mencionada.

Validación de correo electrónico

Dado que por especificación no puede existir más de un usuario con el mismo correo electrónico dentro del sistema, se diseñó una herramienta que recibe una cadena y la compara contra los correos electrónicos de los usuarios registrados. De hallar alguna coincidencia devuelve la cadena "1", en otro caso regresa la cadena "0". De este modo, los programadores de la siguiente capa podrían evitar que un usuario complete su registro si proporciona un correo electrónico registrado previamente.

Validación de Tarjeta de Crédito

Bajo ciertas circunstancias el sistema *Payworks®* permite hacer solamente un cargo por día a una tarjeta de crédito determinada. Considerando que la especificación el sistema para el programa *Coleccionando Sonrisas* indica que pueden existir diversos usuarios registrados con la misma tarjeta de crédito por un lado, y que al registrarse con tarjeta de crédito se lleva a cabo un cargo por concepto de afiliación al programa por el otro, fue necesario diseñar una herramienta que permitiera a los programadores de la capa vista/control evitar que se registraran dos usuarios con el mismo número de tarjeta el mismo día. Así se concretó el diseño de la herramienta de validación, que recibe como parámetro el número de la tarjeta y devuelve falso si la tarjeta existe y se le ha hecho algún cargo en las últimas 24 horas, o bien verdadero si dicho número no está registrado o lo está pero no hay impedimento temporal para efectuar un cargo sobre el mismo.

Donación de puntos

Con el objetivo de cumplir con la especificación del caso de uso descrito en la Sección 2.2.5 se diseñó una herramienta que recibe los siguientes parámetros:

1. Identificador unívoco del donante.
2. Nombre de usuario del destinatario.
3. Cantidad de puntos a donar.

Esta herramienta deberá verificar la consistencia de los datos proporcionados, si los datos son consistentes intentará enviar un correo electrónico a la dirección registrada en el perfil del destinatario. Será necesario conocer el resultado de la ejecución de este proceso desde la aplicación, para tal efecto éste regresará un valor numérico de acuerdo a la siguiente lista:

0: No se logró la transacción o la notificación al destinatario.
1: La transacción fue exitosa.
2: El donante no tiene puntos suficientes.
3: El login proporcionado no se encontró en la base de datos.

De esta manera, el equipo responsable de la capa vista/control podrá elegir el mensaje más adecuado a mostrar según el resultado de la operación.

Búsqueda de CLABE

Para garantizar que no se le permita a un usuario registrarse con la misma CLABE o tarjeta de débito se ha diseñado una herramienta que recibe una secuencia de 18 dígitos decimales, la cual verifica que esta secuencia no esté registrada como número de tarjeta de débito o clave bancaria estandarizada dentro de la base de datos. El resultado se entrega como un valor booleano.

Inserción de pirámides

Considerando la existencia de un formulario para capturar datos de una pirámide nueva, se diseñó una herramienta que, recibiendo como entrada los siguientes atributos (subconjunto de los descritos en el Cuadro 3.6):

- nombre de la pirámide
- saludo

- imagen

crea el registro correspondiente en la Tabla `Arbol` tomando como valores predeterminados para el resto de los atributos los siguientes:

id: El siguiente en la secuencia.
altura: 0.
completo: falso.
raiz: nulo.
altura_máxima: 12.
último_corte: nulo.

4.3.3 Generación de archivos de cobranza (crédito)

Dado que los datos que contienen estos archivos son muy sensibles y considerando que los datos, aunque encriptados, estarían almacenados en un servidor y tendrían que ser extraídos del mismo y a su vez copiados en algún medio temporal para que el operador técnico pudiera enviarlos *Payworks®* la *FPN* solicitó explícitamente que esta herramienta cumpliera con tres características esenciales:

- Que la descarga no proviniera de la página web del programa *Coleccionando Sonrisas*.
- Que sólo pudieran usarla usuario con rol de `Operador Técnico`.
- Que el intercambio de datos entre el servidor y el dispositivo usado por el usuario fuera seguro.

Los requisitos mencionados dieron como resultado un diseño muy peculiar que consta de una aplicación que se ejecutará en el dispositivo del usuario la cual se conectará al servidor, usando el protocolo *Secure Shell*[8] con el método de autenticación por llave pública RSA, intercambiará toda la información necesaria para iniciar la sesión del operador técnico en el sistema, solicitará la contraseña del supervisor del patronato e iniciará así la descarga del archivo con los datos decodificados.

El funcionamiento de esta aplicación se puede describir con el diagrama de flujo mostrado en la Figura 4.3.

[8]Se profundizará sobre este protocolo en la Sección 5.4.2

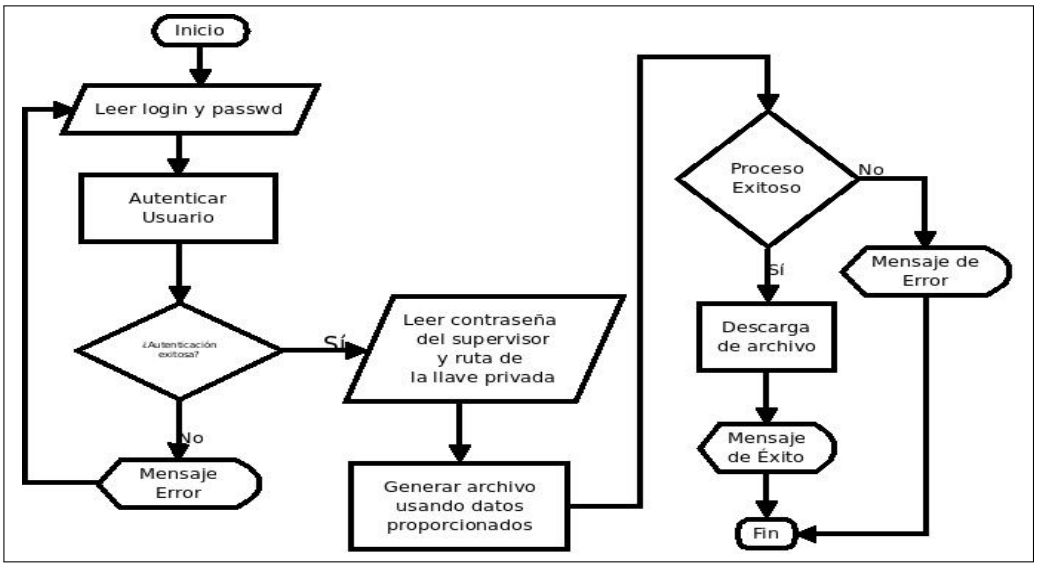


Figura 4.3: Diagrama de flujo del generador de archivos de cobranza a crédito.

4.3.4 Herramientas para procesos asistidos

El diseño de estas herramientas tiene la finalidad de facilitar a los usuarios con rol de `Administrador` llevar a cabo las actividades que tendrán designadas dentro del programa *Coleccionando Sonrisas*. Las dos más relevantes dentro del contexto de este trabajo son: la distribución mensual de puntos y la generación (y proceso) de archivos de intercambio de información (descritos en la Sección 2.4). Considerando el perfil profesional de estos usuarios y con el objetivo de minimizar costos y tiempos de entrega se gestaron las herramientas basadas en línea de comandos que se describen a lo largo de esta sección.

Graficador de pirámides

Tal y como se describe en la Sección 2.4, antes de comenzar una distribución de puntos el administrador deberá generar una representación gráfica para cada una de las pirámides dentro del sistema y ponerla a disposición de el o los operadores técnicos.

Se diseñó una herramienta que recorrerá la lista de árboles registrados en la base de datos y para cada uno generará el código correspondiente en el lenguaje de

definición de gráficas DOT[7]. Una vez generado el archivo, la la aplicación deberá procesarlo usando un software de conversión para generar un archivo en formato de gráficos vectoriales redimensionables[26], SVG[9].

El formato SVG es soportado por navegadores web y múltiples visores de imágenes, por lo que el diseño no supone un problema para que los operadores técnicos puedan visualizar las representaciones gráficas de las pirámides.

Distribución de puntos

Esta herramienta recibirá como parámetro la ruta relativa de un archivo que contendrá un identificador de usuario en cada línea. Dicho archivo será interpretado como la lista de usuarios morosos, esto es, a quienes por alguna razón no se les deben agregar puntos durante la ejecución del programa.

En primera instancia, se recorrerá la lista de donantes registrados en la base de datos, en cada paso se actualizará el registro correspondiente en la Tabla `puntos` con el valor 2 en la columna `estado_id` si dicho usuario aparece en la lista de morosos y con el de 4 en otro caso[10]. Además, a cada uno de los donadores activos se le abonarán los 10 puntos correspondientes del mes. Este proceso se puede apreciar con mayor claridad en el Algoritmo 4.

input : Lista LM de usuarios morosos
input : Lista US de usuarios registrados
for $(usuario \in US)$ **do**
  if $(u \notin LM)$ **then**
    $u.estado\_id = 2;$
    $u.puntos+ = 10;$
  else
    $u.estado\_id = 4;$
  end
end

Algoritmo 4: Caracterización del proceso de distribución de puntos que ejecutará el administrador.

Una vez suspendidos todos los usuarios morosos, activados los que no los sean

[9] SVG son las siglas en inglés de *Scalable Vector Graphics*
[10] El valor 2 representa al estado suspendido mientras que el 4 al estado normal.

y abonados los 10 puntos correspondientes del mes; se ejecutará el algoritmo de distribución de puntos (Algoritmo 2) para cada uno de los árboles registrados dentro del sistema.

Proceso de respuestas de afiliación

Como se describió en la Sección 2.2.1 el registro de donantes con tarjeta de débito o CLABE trae consigo la generación de un archivo de comunicación generado por el sistema *Payworks®*. Este diseño proveerá al administrador una herramienta con la que podrá reportar al operador técnico el resultado del trámite de afiliación de las cuentas de los donantes además del archivo de cobranza correspondiente.

Tomará como parámetro las rutas relativas de los archivos de respuesta a procesar. Para cada registro de cada uno de los archivos la aplicación verificará el resultado de la transacción. Actualizará el registro correspondiente en `alta` cambiando el estado a "preproceso" si la afiliación fue completada exitosamente o a "rechazado" si la respuesta de *Payworks®* fue negativa.

Una vez completada la lectura de los archivos, la herramienta buscará todos los registros de alta en estado "preproceso" o "pendiente" y generará un archivo de cobranza por afiliación (con extensión .PAG). Habiendo generado el archivo los registros de alta completados serán actualizados al estado "proceso".

La salida de este programa consistirá en la ruta relativa del archivo .PAG que se generó y la lista de id de usuario cuya afiliación fue rechazada. El administrador deberá enviar ambas cosas al operador técnico para que lleve acabo las tareas pertinentes.

Generación de archivos de cobranza (débito)

Antes de describir el diseño de esta aplicación es necesario recapitular un poco acerca del tema de cobranza a tarjetas de crédito y débito. Como hemos observado en secciones anteriores es posible distinguir entre dos tipos de cobranza a cuentas de ahorro de los donantes, a saber, cargos mensuales y cargos por afiliación.

Dadas las similitudes que existen entre ambos procedimientos, diferiremos solamente la definición de la lista de donantes (contribuyentes) a recorrer y en los párrafos

subsecuentes se describirá el proceso unificado para generar los archivos .PAG.

Cargo por Afiliación

En la sección siguiente se hará evidente la posibilidad de que un registro de alta quede en estado pendiente de pago, caso en el cual se deberá agregar al próximo archivo de cobranza por afiliación generado. Es dable entonces, para facilitar la comprensión del funcionamiento de esta herramienta definir como contribuyentes a aquellos donantes que en un momento dado cumplen con las siguientes condiciones:

- Estar registrados con tarjeta de débito o CLABE.
- Estar asociados a un registro de alta en estado ”proceso” o ”pendiente”.
- Estar activos.

Cargo Mensual

A diferencia de los cargos por afiliación, la lista de contribuyentes para los cargos mensuales no puede incluir donadores que aún no completan su incorporación al sistema. En consecuencia, los contribuyentes para cargos mensuales deberán cumplir con las siguientes condiciones:

- Estar registrados con tarjeta de débito o CLABE.
- No estar asociados con algún registro de alta.
- Estar activos.

Proceso general

La aplicación deberá recorrer dos veces la lista de contribuyentes, en el primer recorrido contará y calculará el número de cargos por realizar y el monto total por los mismos. Escribiendo en un archivo temporal el encabezado correspondiente en el formato compatible con *Payworks®*.

El objetivo del segundo ciclo es generar, para cada contribuyente, una cadena compatible con el contenido esperado por *Payworks®*, la cual se añadirá al archivo temporal mencionado en el párrafo anterior.

Una vez contando con todos los registros de cobranza almacenados temporalmente, la aplicación deberá enviar un correo electrónico destinado a los operadores técnicos adjuntando el archivo (con extensión .PAG) que se generó, eliminándolo permanentemente del dispositivo de almacenamiento donde se encuentre después del envío exitoso del mensaje.

La salida generada por esta aplicación consistirá en mensajes de texto plano indicando[11] el número de registros de cobranza generados y el estado del envío del correo electrónico. En caso de ocurrir algún error durante el proceso, éste se verá reflejado en los mensajes de salida y se deberá ejecutar de nuevo el proceso.

Proceso de respuestas de cobranza

La aplicación de proceso de respuestas de cobranza a cuentas de ahorro tiene como objetivo facilitar y automatizar las tareas necesarias después de que se recibe un pago exitoso.

El diseño de esta herramienta considera 5 tareas fundamentales:

- Separar los registros en archivos distintos según el tipo de cargo[12].
- Procesar archivos con respuestas de cobranza por afiliación.
- Generar el reporte cargos rechazados.
- Generar la lista de identificadores de usuarios cuyos cargos fueron rechazados.
- Generar la lista de identificadores de usuarios cuyos cargos fueron aceptados.

El programa deberá recibir como parámetro las rutas relativas de los archivos a procesar y alguna palabra clave que identifique la tarea que debe realizarse.

Al generar de listas de identificadores o reportes, sólo se beberán leer los archivos, identificar (para cada registro) si el cargo fue exitoso o no y generar el mensaje adecuado en la salida estándar. Dicho mensaje puede ser un identificador de usuario o bien, en el caso del reporte de cargos rechazados, una cadena que contenga los siguientes datos:

- Nombre del archivo en el que se encuentra el registro.
- Fecha en la que se procesó el archivo.

[11] Que se mostrarán en la consola de comandos.

[12] Considerando que dentro de los archivos de respuesta de *Payworks®* no existe distinción alguna entre los diferentes conceptos de cargo mencionados en la sección anterior.

- Identificador del donador.
- Clave del motivo de rechazo del cargo.

Por otro lado, si se trata del proceso de respuestas de cobranza por afiliación, la aplicación deberá leer todas las líneas de todos los archivos y localizar para cada una el registro correspondiente en `alta`. Para cada alta encontrada deberá actualizar el estado a ”postproceso” si el cargo fue exitoso o a ”pendiente” en otro caso.

Después de actualizar todas las altas relacionadas con los archivos procesados, se deberán recorrer los registros con estado ”postproceso” y para cada uno:

1. Vincular al donador a la pirámide referida.
2. Eliminar permanentemente el registro en `alta`.

La aplicación deberá reportar al administrador si alguno de los donantes no se pudo vincular a la pirámide correspondiente. En otro caso no reportará salida alguna.

Capítulo 5

Requerimientos Técnicos

En este capítulo se describirán brevemente las herramientas, lenguajes de programación, manejador de bases de datos y el servidor *web* que fueron usados para implementar las aplicaciones cuyo diseño se describe en el Capítulo 4.

5.1 Plataforma

Esta sección pretende aportar información general acerca del sistema operativo, servidor *web* y manejador de base de datos sobre los que está montado el sistema *Coleccionando Sonrisas*[1].

5.1.1 CentOS GNU/Linux

CEntOS es una distribución gratuita del sistema operativo GNU/Linux. Fue creada en 2004, es de código abierto bajo la licencia GNU-GPL[2] y "funcionalmente compatible con *RedHat Enterprise Linux*"[22]. Su gestión de paquetes RPM[1] permite que la instalación, actualización, configuración y mantenimiento de paquetes sea ágil y segura.

Lo anterior, aunado al amplio catálogo de optimizaciones y mejoras implementadas sobre su núcleo de sistema, hace de CEntOS una de las distribuciones gratuitas

[1]En este capítulo usaremos el término "sistema *Coleccionando Sonrisas*" para referirnos al sistema que se elaboró para el programa *Coleccionando Sonrisas*

[2]La licencia GPL es la licencia bajo la cual se distribuye el software libre[5]

de GNU/Linux más confiables dentro del mundo de los servidores. Motivo por el cual se eligió como sistema operativo para el *hardware* en el que se ejecuta el sistema *Coleccionando Sonrisas*.

5.1.2 Servidor *web* Apache

La primera versión del servidor HTTP[3] Apache fue liberada en abril de 1995, basado en el servidor más popular del momento: el HTTPD del NCSA[4] resulto ser todo un éxito.[21]

Mantenido por un grupo de notables voluntarios, este servidor web se ha vuelto una herramienta confiable, segura, eficiente y extensible. De más está decir que según las encuestas es el número uno en Internet.[20]

Al ser de distribución gratuita y tener tan amplias cualidades, se volvió la opción ideal para el sistema *Coleccionando Sonrisas*.

5.1.3 Manejador MySQL

Es un manejador de bases de datos que soporta el lenguaje de consultas estructurado[5]. Se ha vuelto extremadamente popular en el desarrollo de aplicaciones *web* gracias a su fiabilidad, alto desempeño y facilidad de configuración, administración y uso.[14]

Al no contar con revisores de integridad referencial ni manejo estricto de transacciones en su configuración por defecto, sus mínimos requerimientos de *hardware* y alto desempeño en conjunción con el amplio catálogo de herramientas de conexión diseñadas para el lenguaje PHP, hacen de MySQL la opción más adecuada para mantener la base de datos del sistema *Coleccionando Sonrisas*.

5.2 Base de datos

A continuación se discutirán dos temas sumamente relacionados con la implementación de la capa de persistencia de datos del sistema *Coleccionando Sonrisas*.

[3]HTTP son las siglas en inglés de *Hypertext Transfer Protocol*[9]
[4]National Center Of Supercomputing Applications
[5]También conocido como SQL por las siglas en inglés de *Structured Query Language*[19]

Con el objetivo de minimizar la carga de operaciones que se ejecutan en la capa de aplicación y, al mismo tiempo, mantener un diseño funcional y transparente para los desarrolladores de las capas superiores algunos de los requerimientos de operación del sistema se implementaron directamente en el nivel de datos mediante procedimientos almacenados (*stored procedures*) y *trigger*.

5.2.1 Procedimientos Almacenados

Los procediminetos almacenados secuencias de comandos, que pueden o no estar parametrizadas, que el manejador de bases de datos puede ejecutar sobre demanda. Al tener acceso inmediato a los datos, la lógica de negocio implementada con *stored procedures* puede ser más ágil y eficiente que la tradicional a través de conexiones desde una capa superior a la capa de datos pues se minimiza el costo de intercambiar tanto las solicitudes como las respuestas entre las capas.

El manejador MySQL soporta procedimientos almacenados de dos tipos, a saber: funciones y procedimientos[13]. En el caso particular del sistema *Coleccionando Sonrisas* se usaron procedimientos almacenados para implementar la caducidad de los puntos. Todos los días, un software de calendarización ejecuta una consulta a la base de datos llamando a un procedimiento almacenado, el cual invalida los puntos que tienen más de un año de vigencia.

5.2.2 *Trigger*

Otra herramienta de los manejadores de bases de datos que simplifica la implementación de capas superiores son los *trigger*. A grandes rasgos, un trigger es un proceso que se ejecuta cuando ocurre algún evento que altere los datos almacenados en cierta tabla de una base de datos.[13]

La base de datos del sistema *Coleccionando Sonrisas* utiliza un *trigger* para actualizar los datos (como la altura, el número de nodos, etcétera.) de la Tabla `arbol` cada que ocurre un cambio en la Tabla `puntos`. Asimismo, se utiliza otro *trigger* para actualizar la vigencia de los puntos cada que la Tabla correspondiente presenta una alteración en este campo.

5.3 Lenguajes de Programación

Los lenguajes ocupados por el sustentante para implementar su parte del trabajo fueron Bash, Java y PHP. A continuación se describen a grandes rasgos las características y motivos por los que se eligieron.

5.3.1 Java

Java es un lenguaje de alto nivel, orientado a objetos, sumamente flexible y multiplataforma. Su gran versatilidad en términos del tipo de aplicaciones que se pueden desarrollar en el mismo lo han convertido en uno de los lenguajes de programación más usados a lo largo de la historia,[15].

Java surgió en 1990 como un lenguaje cuyo propósito era controlar microprocesadores instalados en aparatos electrónicos como tostadoras, decodificadores de televisión de paga y computadoras de bolsillo,[10].

Una parte de la aplicación de generación de archivos de cobranza a crédito fue implementada en Java puesto que esta debía ejecutarse en el equipo del operador técnico y no en el servidor. Las herramientas SWING[10] para el diseño de interfaces gráficas y la capacidad de ejecutar comandos externos[10] fueron el motivo por el cual se eligió este lenguaje para desarrollar la aplicación mencionada.

5.3.2 PHP

PHP surgió en 1995 como un pequeño script elaborado por Rasmus Lerdorf en Perl/CGI para contar el número de visitantes que leían su currículum en línea. Este script comenzó a ganar popularidad al ser innovador para la época y poco a poco se fue convirtiendo en un conjunto de herramientas, siendo una de ellas la capacidad de convertir datos capturados en una forma HTML en variables simbólicas, apodado *Personal Home Page*,[8].

PHP ha estado entre los 10 lenguajes más populares del mundo desde 1999 y ha llegado incluso al tercer lugar en el año 2010,[25].

Gracias a su amplia penetración en el desarrollo de aplicaciones web, el equipo de desarrollo del proyecto *Coleccionando Sonrisas* decidió comenzar la implementación

de la aplicación en este lenguaje. Al iniciar la participación del autor de este texto se actualizó la versión del lenguaje de 4.0 a 5.3 para aprovechar al máximo los beneficios de la programación orientada a objetos.

5.4 Software de Apoyo

Ni el desarrollo ni la puesta en marcha de este proyecto se habrían logrado de no contar programas creados, en su mayoría, por terceros. En esta sección se hará una breve descripción de cada una de las herramientas que hicieron posible la implementación del sistema *Coleccionando Sonrisas* dando el crédito necesario a sus creadores.

5.4.1 Bash

Siendo el intérprete de comandos predeterminado para la mayoría de las distribuciones de GNU/Linux[18], fue necesario usar Bash para instalar, configurar y poner en marcha todo el sistema.

Además, para completar la implementación de la aplicación diseñada en la Sección 4.3.3 se creó un programa en *Unix Shell*[6] para generar el archivo que se descarga después al cliente.

5.4.2 Secure Shell

Mejor conocido como SSH, es un protocolo de red que permite a dos equipos:

- transferir datos,
- autenticar usuarios (iniciar sesiones) y
- ejecutar comandos remotamente

de modo tal que nadie, que pueda capturar los datos durante su tránsito, pueda conocer la información que se está intercambiando,[2].

Gracias a que la mayoría de las distribuciones de GNU/Linux cuentan con servidor SSH, la mayor parte de la instalación, configuración e implantación del sistema

[6]Aunque es más común escuchar este término para referirse al intérprete de comandos, se trata también de un lenguaje de programación[18]

en el servidor de la *FPN* se llevó a cabo haciendo conexiones remotas al mismo a través de este protocolo.

Por otra parte, se usó este protocolo para proteger la transferencia de del archivo que genera la aplicación diseñada en la Sección 4.3.3 a través de del programa *Secure Copy* (que es una implementación del protocolo homónimo) generalmente incluido cualquier software que implementa SSH.

5.4.3 Mercurial

El control de versiones es una práctica indispensable en el ámbito del desarrollo de software, sin embargo, llevarla a cabo a través de un software automatizado tiene ventajas considerables: el poder revertir los cambios hasta un punto específico si se han cometido errores, la capacidad de modificar simultáneamente el mismo archivo por varias personas, tener una historia detallada de las modificaciones y los responsables de las mismas, etcétera.

Mercurial es un gestor de versiones eficaz, distribuido, multiplataforma, versátil y fácil de usar. Su desarrollo está inspirado en Monotone[12] y fue candidato junto con Git para ser usado por la comunidad de desarrolladores que mantienen el proyecto del kernel de GNU/Linux,[16].

Al estar involucrados varios programadores y diseñadores web trabajando en distintos lugares y momentos sobre los mismos archivos fue de gran provecho mantener, al menos durante la participación del sustentante en la implementación del sistema, un repositorio de código fuente administrado por Mercurial.

5.4.4 Calendarización en GNU/Linux

El programa de calendarización para la mayoría de las distribuciones de GNU/Linux llamado **mCron** está basado en el software incorporado en las implementaciones de Unix de AT&T y Berkeley. Dicho software fue documentado por Paul Vixie y rediseñado completamente por Dale Mellor,[11].

Usando **mCron** es posible hacer que ciertos comandos (o programas en *Unix Shell*) sean ejecutados periódicamente especificando la fecha, hora e intervalo de repetición.

La calendarización de la llamada al procedimiento almacenado de caducidad de puntos descrito en la Sección 5.2.1 se logró gracias al programa mCron incluido dentro de la versión de CentOS instalada en el servidor de la *FPN*.

Conclusiones

Resultados

El sistema implementado para el programa *Coleccionando Sonrisas* sigue activo al día 4 de junio de 2014 con más de 750 donadores registrados. Se han registrado más de 800 intercambios de puntos por artículos y actividades y existen 34 pirámides distintas, muchas de ellas registradas por convenios de colaboración con empresas privadas.

Actualmente, el ingreso generado por el programa *Coleccionando Sonrisas* para la *FPN* es mayor al costo de su mantenimiento, administración y operación, motivo por el cual el patronato de la fundación ha decidido extender su aplicación de manera indefinida.

Lo anterior ha motivado a los dirigentes del programa a comprar espacios publicitarios en radiodifusoras con cobertura en toda la República así como en diversos medios impresos de circulación nacional.

Experiencia

La experiencia en el desarrollo de este proyecto fue muy útil para corroborar que la formación que se adquiere en la licenciatura en Ciencias de la Computación permite al profesional desenvolverse en proyectos de desarrollo de software a gran escala con versatilidad, agilidad y contundencia. Sin embargo, en el caso particular del sustentante expuso algunas debilidades en términos de presupuesto, gestión de proyectos y cálculo de tiempos de entrega.

Trabajo a futuro

Diversas circunstancias impidieron que el proyecto se desarrollara de la mejor manera posible desde el punto de vista del autor de este reporte. Finalizaremos el texto con una serie de observaciones y propuestas detalladas sobre las mejoras que pueden llevarse a cabo al software desarrollado.

Modularización

A pesar de que el diseño de las herramientas de acceso a datos desde la aplicación incluye clases distintas para representar a cada una de las entidades están involucradas en el sistema, la implementación actual de la capa de acceso a datos está concentrada en una sola clase debido a la falta de experiencia de los desarrolladores de las capas superiores en el uso de orientación a objetos dentro del lenguaje PHP. Sería una mejora significativa implementar o implantar un software de mapeo Objeto-Relacional para facilitar la manipulación de los datos asociados a cada entidad desde las capas superiores y orientar la arquitectura de la aplicación hacia el patrón modelo Vista/Controlador.

Encripción de datos

Aunque el algoritmo de encripción usado para cifrar los datos bancarios de los usuarios fue una creación original, podría ser vulnerable ante ataques por fuerza bruta. A pesar de que la complejidad de romper este esquema criptográfico no es baja considerando que el número de posibles contraseñas sería 16^{10}, rediseñar el algoritmo o utilizar alguno ya existente podría mejorar notablemente la seguridad de la información.

Por otro lado, es ampliamente recomendable incorporar algún algoritmo de derivación de claves con el fin de aumentar la seguridad y acercar más la implementación del sistema a las buenas prácticas de diseño de sistemas criptográficos.

Distribución mensual de puntos

Después de analizar con detalle el crecimiento de los árboles dentro del sistema se descubrió que éstos no tienden al balance, lo que hace al algoritmo de distribución de puntos más costoso de lo que se había contemplado.

Una posible solución sería almacenar dentro de la base de datos la cantidad de puntos que cada usuario debería recibir según el subárbol del cual es raíz. Sin embargo, esto implicaría que por cada nuevo donante que se registre se tendría que recalcular el número de puntos a recibir para todos sus ancestros, lo que podría tener un efecto de retraso en el proceso de registro del usuario para árboles muy grandes.

Otra opción sería hacer una investigación más profunda sobre algoritmos de recorrido de árboles binarios esperando que existiera alguno descubierto previamente que puede usarse o adaptarse a para su aplicación en el programa *Coleccionando Sonrisas*.

Bibliografía

[1] Edward C. Bailey. *Maximum RPM: Taking the Red Hat package manager to the limit*. Red Hat Software, 1997.

[2] Daniel J. Barrett y Richard E. Silverman. *SSH, the Secure Shell: The Definitive Guide*. O'Reilly & Associates, Inc., 2001.

[3] Consejo Nacional de Evaluación de la Política de Desarrollo Social. *Metodología para la medición multidimensional de la pobreza en México*. 2009. URL http://web.coneval.gob.mx/Informes/Coordinacion/INFORMES_Y_PUBLICACIONES_PDF/Metodologia_Multidimensional_web.pdf.

[4] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, y Clifford Stein. *Introduction to Algorithms*. McGraw-Hill/MIT Press, 2009.

[5] Free Software Foundation. https://www.gnu.org/licenses/gpl-3.0.html. (Recuperado el 7 de abril de 2014).

[6] José J. Galaviz Casas. *Introducción a la Teoría de Códigos, Teoría de la Información y Criptografía*. UNAM.

[7] Emden R. Gansner, Eleftherios Koutsofios, y Stephen North. *Drawing graphs with dot*. 2010. URL http://www.graphviz.org/pdf/dotguide.pdf.

[8] W. Jason Gilmore. *Beginning PHP and MySQL : from novice to professional*. Apress, 2010.

[9] Network Working Group. http://tools.ietf.org/html/rfc2616. (Recuperado el 7 de abril de 2014).

[10] Clark S. Lindsey, Johnny S. Tolliver, y Thomas Lindblad. *JavaTech: Introduction to Scientific and Technical Computing with Java*. Cambridge University Press, 2005.

[11] Dale Mellor. http://www.gnu.org/software/mcron/design.html#section1. (Recuperado el 7 de abril de 2014).

[12] Monotone.ca. http://www.monotone.ca. (Recuperado el 7 de abril de 2014).

[13] Joel Murach. *Murach's MySQL*. Mike Murach & Associates Inc., 2012.

[14] MySQL.com. http://www.mysql.com/about/. (Recuperado el 7 de marzo de 2014).

[15] Oracle.com. http://www.java.com/en/about/. (Recuperado el 10 de marzo de 2014).

[16] Bryan O'Sullivan. *Mercurial: The Definitive Guide*. O'Reilly Media, Inc., 2009.

[17] Nélida B. Perona y Graciela I. Rocchi. *Vulnerabilidad y Exclusión social. Una propuesta metodológica para el estudio de las condiciones de vida de los hogares. Kairos, Revista de Temas Sociales*, (8), 2000. URL http://www2.fices.unsl.edu.ar/~kairos/k08-08.htm.

[18] Chet Ramey y Brian Fox. *Bash Reference Manual*. Free Software Foundation, 2010.

[19] Abraham Silberschatz, Henry F. Korth, y S. Sudarshan. *Database System Concepts*. McGraw-Hill, 2010.

[20] *Apache.org*. http://httpd.apache.org/. (Recuperado el 4 de marzo de 2014).

[21] *Apache.org*. http://httpd.apache.org/about_apache.html. (Recuperado el 4 de marzo de 2014).

[22] *CEntOS.org*. http://www.centos.org/about/. (Recuperado el 25 de febrero de 2014).

[23] *Fundación para la Protección de la Niñez I. A. P.* http://www.infanciamexico.org/quienes_somos/. (Recuperado el 7 de abril de 2014).

[24] *Infoniñez.mx.* http://www.infoninez.mx/dashboard/app/webroot/stock/profile/lacmex.pdf. (Recuperado el 5 de octubre de 2012).

[25] *TIOBE Software.* http://www.tiobe.com/index.php/content/paperinfo/tpci/php.html. (Recuperado el 3 de abril de 2014).

[26] *W3C.* http://www.w3.org/graphics/svg/about.html. (Recuperado el 29 de octubre de 2004).