Un proyecto de construcción de software de aplicación para ser desarrollado en los primeros años de una carrera de Ciencias de Computación: Un sistema de cruzamiento de encuestas

Jorge Aguirre

Área Computación FCEFQyN UNRC, Dto. Computación FCEyN UBA

Ruta 8 Km. 603 UNRC

Email: jaguirre@reinfo.uba.ar

Ricardo Medel

Área Computación FCEFQyN UNRC

Ruta 8 Km. 603 UNRC

Email: rhmedel@exa.unrc.edu.ar

Resumen:

Se presenta un proyecto para ser realizado en un taller o como complemento de una de las primeras asignaturas de programación. Así mismo se presenta la experiencia realizada en la UNRC donde este proyecto se aplicó en el segundo año de las carreras de Ciencias de la Computación. Se ha seleccionado el proyecto de manera que conduzca a utilizar las técnicas de construcción de programas correctos vistas con anterioridad y distintos algoritmos de manejo de estructuras de datos simples, principalmente listas y algunas de sus restricciones, como las pilas. Se ha buscado que el problema a resolver pertenezca a un dominio de aplicación concreto y que dentro de él sea relevante para que induzca una mayor motivación, aunque permita una solución simple e implementable con los conocimientos adquiridos por los alumnos al momento de su desarrollo. El tema seleccionado es el cruzamiento de encuestas. En el proyecto se plantea la construcción de un programa que permita obtener un cuadro de cruzamiento a partir de una especificación del mismo. Para la especificación del cuadro se propone un lenguaje declarativo, que permite describir el formato del cuadro, las condiciones cuya frecuencia de ocurrencia quiera contarse y el subdominio encuestal sobre el que se desea realizar el proceso. El lenguaje de especificación está pensado para permitir que pueda ser reconocido sin necesidad de conocer técnicas de análisis sintáctico.

1 Introducción

Estamos convencidos de la necesidad de que los alumnos de carreras de Ciencias de la Computación aborden en distintas etapas de su carrera actividades de concreción de proyectos de desarrollo de software que abarquen todas las etapas del ciclo de desarrollo. Esta actividad organizada en talleres es utilizada tradicionalmente en otras áreas que también requieren el desarrollo de la capacidad de diseño del estudiante y el desarrollo de su capacidad instrumental Un ejemplo paradigmático del uso de esta modalidad lo es la enseñanza de la Arquitectura.

Los proyectos deben ser seleccionados de manera tal que su solución abarque la mayor cantidad posible de los conocimientos adquiridos en las asignaturas ya cursadas. En este sentido se diferencian de la realización de trabajos prácticos de implementación concreta. Aquí la generalidad del problema a resolver exige la realización de un considerable trabajo de análisis y de síntesis. El primer problema es encontrar la modelización que permita utilizar los resultados generales que el alumno ha estudiado previamente, sin que el contexto sugiera directamente apelar a determinado conocimiento o usar la técnica objeto de estudio en ese momento, como sucede en la habitualmente en la realización de trabajos prácticos. Pensamos que esta capacidad sólo puede desarrollarse en la práctica de la resolución de problemas de complejidad gradual. El docente debe guiar, criticar las soluciones y mostrar alternativas mejores, pero nunca desplazar al alumno del papel protagónico en el desarrollo del proyecto.

Estos trabajos crean también el marco adecuado para el aprendizaje asistido del trabajo en grupo. En ellos puede abordarse la problemática que crea la modularización de un proyecto, el desarrollo paralelo de distintas actividades llevadas a cabo por varias personas y la coordinación del trabajo individual.

Así mismo consideramos importante que durante la realización de los proyectos se desarrolle la capacidad de elaborar documentos, cubriendo tanto reportes técnicos como documentación interna del sistema, destinada a su mantenimiento, y manuales didácticos destinados al usuario final.

La modalidad de talleres o laboratorios [Arr96] conforma el marco ideal para el desarrollo de este tipo de actividades, independizando la labor proyectual de las asignaturas particulares, jerarquizando la actividad y dándole un espacio propio en la organización curricular. No obstante, en las carreras cuyo plan de estudios no incluya talleres, estos proyectos pueden -creemos que deben- llevarse a cabo dentro de las asignaturas afines.

En este artículo se presenta un proyecto de construcción de un producto de software para dar solución a un problema específico. El proyecto fue diseñado para ser realizado por alumnos de los primeros años de una carrera de Ciencias de la Computación. Puede ser llevado adelante en un taller o laboratorio, o como complemento de una de las primeras asignaturas de programación.

El proyecto aquí descripto se utilizó en el tercer cuatrimestre de las carreras de Ciencias de la Computación de la Universidad Nacional de Río Cuarto, durante la segunda materia de programación y algorítmica -"Programación Avanzada"-. Proyectos similares fueron usados por uno de los autores en la cátedra de Taller II de la Licenciatura en Informática de ESLAI en 1987 y en 1991.

2 El proyecto

2.1 Objetivos

- Capacidad de utilizar los conocimientos previamente adquiridos para abordar la solución de un problema. Fundamentalmente: Esquemas para la construcción de algoritmos correctos, arreglos y listas y sus variantes (1).

- Capacidad de diseñar soluciones para problemas de cierta complejidad y llevar adelante su implementación.

- Habilidad en el uso del ambiente de programación utilizado hasta el momento.

- Capacidad de elaborar documentos simples.

- Capacidad de trabajo en grupo.

(1) En nuestro caso particular, dado que la asignatura "Programación Avanzada" incluye nociones sobre el uso de Autómatas Finitos y de definición de lenguajes usando BFN, se agregan estos temas.

2.2 Introducción al problema

El resultado final del proyecto debe ser un producto de software que permita obtener cuadros de cruzamientos a partir de los datos de una encuesta. Así, por ejemplo, dados los datos de la Tabla 1, se deberían poder especificar y obtener cuadros como el de la Fig. 1.

sexo	edad	actividad	remuneración
V1	V2	V3	V4
2	35	2	0
2	20	8	20000
2	65	7	30000
1	41	1	2000
1	20	3	550
2	30	8	2000
1	35	6	1000

Tabla 1. Ejemplo de encuesta.

SEXO / REMUNERACIONES / EDAD

	HOMBRES			MUJERES			TOTAL
	0-1000	1001-	TOT	0-1000	1001-	TOT	
EDAD							
0-10	0	0	0	0	0	0	0
11-30	1	0	1	0	2	2	3

30-	1	1	2	1	1	2	4
TOTAL	2	1	3	1	3	4	7

Figura 1. Ejemplo de cuadro.

2.3 El lenguaje de especificación de cuadros.

Se quiere que el lenguaje de especificación de cuadros sea declarativo, suficientemente simple como para poder reconocer sus cadenas sin necesidad de tener conocimientos previos de Análisis Sintáctico y que permita declarar de manera simple y comprensible el formato deseado.

En busca de estas condiciones se utilizó una simplificación del lenguaje utilizado por uno de los autores en un sistema de procesamiento de encuestas desarrollado para la OIT [Agui84]. El lenguaje de especificación resultante se define como sigue:

Un cuadro está formado por una secuencia de comandos, debe haber exactamente un comando X, un comando F y un comando Y, la especificación debe terminar con el comando E.

Los comandos y sus significados se listan en la Tabla 2.

Sintaxis	Descripción	Significado
T <literal>	Título	En la fila correspondiente del cuadro se imprimirá el literal especificado.
I <literal>	Impresión	Este comando está destinado a la impresión de una línea de frecuencias, las frecuencias se imprimen a continuación del literal dado.
P <literal>	Pié de página	El literal será impreso como pie de página.
X <lista de condiciones>	Condiciones de columnas	La i-ésima condición de la lista especifica la condición asociada a la i-ésima columna del cuadro.
Y <lista de condiciones>	Condiciones de filas	La i-ésima condición de la lista especifica la condición asociada a la i-ésima fila del cuadro.
F <formato>	Formato	Especifica el formato de impresión de los resultados en una forma similar a las mascaras de impresión usados por varios lenguajes, como BASIC. Está compuesta por caracteres de separación y '#'s, estos últimos indican la posición de los resultados. Los caracteres de separación entre campos podrán ser cualquiera excepto los siguientes: !$&+-*.
G <condición>	Filtro	Hace que se procesen sólo aquellos registros que satisfagan la condición.
E	Fin (End)	Indica la finalización de la especificación del cuadro.

Tabla 2. Comandos del lenguaje de especificación de cuadros.

Cada lista de condiciones es una secuencia de condiciones separadas por comas, donde una condición es una expresión en notación posfija o polaca inversa entre condiciones simples y las operaciones **Y** (conjunción), **O** (disyunción) y **N** (negación)

Una condición simple puede ser una relación entre variables y constantes, o la pertenencia de una variable a un intervalo cerrado. Cada variable Vn hace referencia al n-ésimo campo del registro de la encuesta.

La BNF que define el lenguaje de las condiciones es la siguiente:

<lista de condiciones> ::= <condición>
| <condición>**,** <lista de condiciones>

<condición> ::= <condición simple>
| <condición> **N**
| <condición> <condición> **Y**
| <condición> <condición> **O**

<condición simple> ::= <variable> <rel> <variable>
| <variable> <rel> <constante>
| <variable> **ε (** <constante> **,** <constante> **)**
| *****

<variable> ::= **V**<entero>

El asterisco (*****) representa la condición que siempre evalúa a Verdadero (True).

Con este lenguaje de especificación, y volviendo al ejemplo previo, el cuadro de la Fig. 1 se especificaría tal como se muestra en la Fig. 2.

```
T              SEXO / REMUNERACIONES / EDAD
T
T          HOMBRES            MUJERES         TOTAL
T       0-1000  1001-   TOT  0-1000  1001-   TOT
TEDAD

I0-10

I11-30

I30-

ITOTAL

F         ###    ###  ###   ###    ###  ###    ###

X V1=1, V4 E (0,1000) Y, V1=1, V4 E (1001,32000) Y, V1=1   &
  V1=2, V4 E (0,1000) Y, V1=2, V4 E (1001,32000) Y, V1=2, *

Y V2 E (0,10) , V2 E (11,30) , V2 E (31,150) , *

E
```

Figura 2. Especificación del cuadro de Fig. 1.

2.3 El enunciado del proyecto.

Se describieron los requerimientos del producto a desarrollar y se definió informalmente el lenguaje de especificación de cuadros.

Se pidió que como primer paso se definiera formalmente dicho lenguaje utilizando BNF.

Se debía entregar un producto en funcionamiento, su documentación y la documentación de uso, destinada a un perfil de usuario no informático.

Para lograr dicho objetivo cada grupo debía discutir con la cátedra los avances sucesivos que realizara sobre el diseño y luego sobre su implementación.

Se dijo que los grupos podían hacer propuestas de modificación de la especificaciones dadas, a fin de mejorar el producto.

Se fijaron metas parciales, de forma tal de completar el desarrollo en el término de treinta días.

3 Discusión de las soluciones propuestas

Para la realización del proyecto los alumnos formaron grupos de desarrollo de cuatro personas como máximo. Las soluciones propuestas por los distintos grupos de desarrollo variaron en cuanto a su eficiencia y nivel de abstracción del problema.

En las subsecciones siguientes detallamos algunas de las características más relevantes de las soluciones propuestas.

3.1 Propuestas de agregados al lenguaje de especificación

Casi todos los grupos propusieron agregar al lenguaje de especificación de formato de cuadro algún comando que simplificara o flexibilizara el uso del sistema.

La incorporación más interesante fue la propuesta de agregar una línea que permitiera especificar el tamaño del registro de la encuesta a procesar.

N <número>	Nº de Variables	Indica la cantidad de valores de cada registro de la encuesta.

Tabla 3. Nuevo comando de especificación propuesto por los desarrolladores.

En algunos casos la línea N se proponía como obligatoria, mientras que la mayoría optó por permitir su omisión, en cuyo caso se aplicaba un número por defecto. Un grupo propuso que el primer dato del archivo de la encuesta fuera el número de variables del registro.

De esta forma se le brindó más flexibilidad al sistema, pues permite adaptarlo a distintos archivos de datos.

3.2 Análisis sintáctico

La mayoría de los grupos realizó el chequeo sintáctico del archivo de texto con la especificación del cuadro, leyendo carácter por carácter directamente del archivo. Pocos de los grupos se dieron cuenta de la ventaja de implementar un primer filtro que ocultara detalles de la cadena de entrada, que podían aparecer en varias partes del análisis posterior, tales como la presencia de tabuladores, espacios adicionales, el carácter de continuación y la división de un comando en varias líneas.

Si bien los alumnos aún no tenían conocimientos sobre métodos de análisis

sintáctico de lenguajes independientes del contexto, dada la simplicidad del lenguaje de las condiciones no tuvieron dificultades en encontrar un algoritmo "ad-hoc" para realizar el análisis.

En la mayoría de los casos los grupos implementaron el reconocimiento de los términos usando autómatas traductores, cuya salida consistía en llamados a procedimientos. El análisis de las condiciones fue implementado con el auxilio de una pila.

Para la evaluación de las condiciones todos los grupos utilizaron una máquina pila, pues en ejercicios realizados con anterioridad ya se había usado esta simple arquitectura.

3.3 Almacenamiento de datos y condiciones

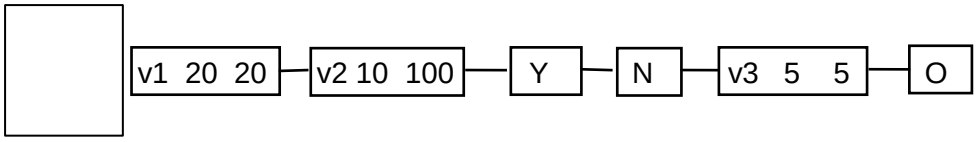
Cabe destacar que todos los grupos almacenaron las condiciones en listas encadenadas. Casi siempre se tenían tres listas: una para las condiciones de las filas, otra para las columnas y una lista de un único elemento para la condición de filtro.

Cada elemento de estas listas era la cabeza de una lista que almacenaba una condición. Por lo general los elementos de estas sublistas eran del tipo registro variante, con dos opciones: operando u operador. Aunque en algunos casos los operadores fueron almacenados como sublistas del último operando.

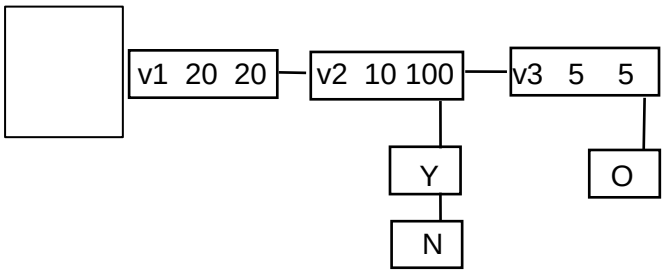
Podemos ver las diferencias de estas dos formas de almacenamiento de condiciones en el siguiente gráfico, en el cual puede observarse también la forma de almacenamiento de la relación de igualdad como si fuese una condición de pertenencia a un intervalo de un único valor.

Condición: V1=20 V2 E (10,100) Y N V3=5 O

Almacenamiento 1:



Almacenamiento 2:



Las condiciones así almacenadas luego eran evaluadas para cada registro de la encuesta por medio del algoritmo clásico, que utiliza una pila para almacenar los resultados intermedios y al encontrar un operador lo aplica al tope, si es unario (N), o a los dos primeros elementos de la pila, si es binario (Y u O).

Finalmente, el cruzamiento de los resultados obtenidos para las condiciones de fila con los resultados de las columnas se realizaba en un arreglo bidimensional de números enteros, a cuyo elemento en posición (i, j) se le sumaba uno sólo si la i-ésima condición de fila y la j-ésima condición de columna habían resultado ambas verdaderas.

Cabe destacar que dos grupos cometieron el grave error de almacenar los resultados de la encuesta en una lista encadenada de registros, copiándolo del archivo original a memoria antes de comenzar el procesamiento de los datos. Esto

produjo una importantísima limitación en el tamaño de las encuestas que se podían procesar con el sistema resultante.

Por último, en lo que respecta al almacenamiento de datos, la mayoría de los grupos desarrolló herramientas para crear los archivos de entrada, a fin de ser utilizadas durante la etapa de testeo y corrección del programa. Dichas herramientas fueron incluidas en el sistema presentado como producto final. Lamentablemente, algunas herramientas utilizaban otro tipo de archivos de datos que los solicitados en la especificación original del sistema (por ejemplo: la encuesta se almacenaba en un archivo de texto), lo que obligaba al usuario a crear los archivos únicamente con dicha herramienta.

3.4 Órdenes de ejecución

En esta subsección analizaremos los distintos tipos de procesamiento realizados sobre los datos de la encuesta, en cuanto a la cantidad de ciclos requeridos para la evaluación de dichos datos.

Las soluciones propuestas pueden ser agrupadas en cinco esquemas simplificados, tal como se muestran en la Fig. 3.

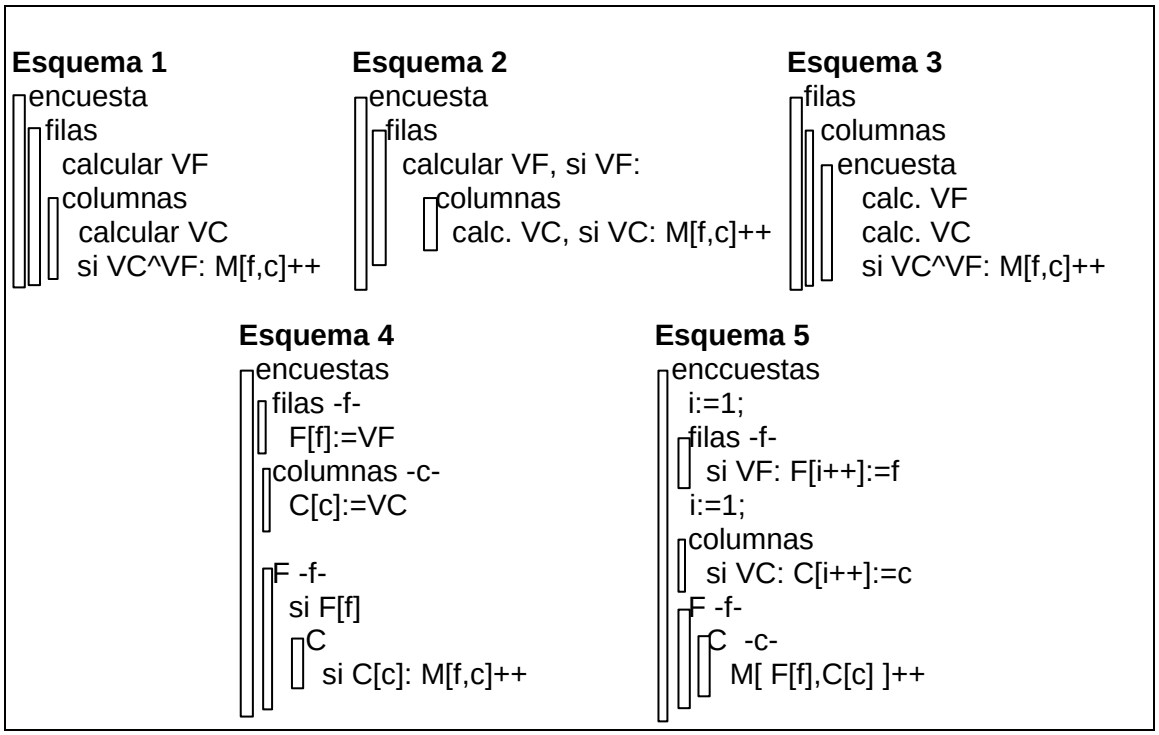


Figura 3. Esquemas de evaluación de encuestas.

Esquema 1:

Por cada registro de la encuesta evalúa todas las condiciones de las filas y por cada fila evalúa todas las condiciones de las columnas. Finalmente a cada elemento de la matriz de frecuencias le asigna el resultado de ambas evaluaciones previas.

Esta fue una solución muy elegida entre los alumnos, probablemente por ser la primera correcta en que pensaban, sin analizar si había otras opciones más eficientes.

Como puede verse, tanto en el peor como en el mejor de los casos, tiene un orden de evaluación O(*n.f.c*), con *n* cantidad de registros de la encuesta, *f* cantidad de filas y *c* cantidad de columnas en el cuadro, suponiendo constante el tiempo de evaluación de las condiciones.

Esquema 2:

Por cada registro de la encuesta evalúa todas las condiciones de la lista X (de

filas). Sólo si la condición resulta verdadera evalúa el mismo registro para todas las condiciones de la lista Y (de columnas).

Esta segunda opción también fue elegida por varios grupos; casi siempre se obtuvo como una mejora del primer esquema. En este esquema el orden de evaluación en el mejor de los casos es sensiblemente menor que en el anterior: O(*n*.*f*), dado que el mejor de los casos es que ningún dato verifique una condición de fila, pudiéndose esperar una mejora importante en los casos reales.

Esquema 3:

Por cada condición de fila y cada condición de columna realiza un ciclo donde las evalúa para todos los registros de la encuesta.

Algunos grupos optaron por esta pésima solución, releyendo todos los datos de la encuesta para cada par de condiciones. En general este esquema estuvo asociado al modelo de almacenamiento donde los datos de la encuesta eran grabados en memoria, al que hiciéramos referencia en la subsección anterior.

Esquema 4:

Para cada registro de la encuesta crea un vector F con los resultados de evaluarlo en cada condición de fila y luego un vector C con los resultados de evaluarlo en cada columna. Finalmente recorre F y en caso de encontrar que alguna condición se cumple asigna a la matriz los resultados en C.

Similar al caso 2, permite un mayor entendimiento del problema, lo que la hace un poco más simple para trabajar y, al estar los vectores implementados como arreglos, más fácil de programar.

Esquema 5:

Para cada registro de la encuesta evalúa cada condición de fila y almacena en un vector F los índices de cada condición que resultó verdadera, lo mismo hace con las condiciones de columnas en un vector C, luego asigna a la matriz de frecuencias M verdadero sólo en los índices almacenados en los vectores F y C.

Esta es la opción más eficiente, pues sólo en el peor caso recorre toda la matriz de cruzamiento. Aunque no fue propuesta originalmente por ningún grupo de desarrollo, se sugirió como una opción válida en algunas clases de consulta, por lo que en algunos casos fue adoptada e implementada.

4 Conclusiones

El proyecto despertó un considerable interés entre los alumnos a los que se les propuso. Sin duda en este sentido resultó importante las características de realidad del problema planteado y la posibilidad de uso del producto obtenido.

La variedad de soluciones correctas a que se podía arribar permitió realizar una interesante discusión sobre los distintos diseños.

Se pudieron poner en práctica los conocimientos previos sobre estructuras de datos simples, sobre definición sintáctica de lenguajes, sobre comparación de algoritmos y sobre el uso de Autómatas Finitos para el reconocimiento de lenguajes regulares.

Se manifestó una importante dificultad para la escritura de documentos. Se realizaron algunos progresos durante el curso del proyecto pero resultó claro que se debía continuar insistiendo en el desarrollo de esa capacidad. Al respecto creemos que las primeras experiencias en este sentido deben ser muy pautadas y que tendríamos que haber detallado más los requerimientos en este aspecto.

5 Referencias.

[Agui84] Aguirre, J. et al. **"Sistema de Tabulación de encuestas"**, Anales de las XIV Jornadas Argentinas de Informática e Investigación Operativa y II CLAIO, Buenos Aires, 1984.

[Arr96] Arroyo, M. y Aguirre, J. **"Una Experiencia en la Enseñanza de Teoría de Lenguajes y Compiladores"**, Anales del 2º Congreso Argentino de Cs. de la Computación, 4º Ateneo de Profesores Universitarios de Computación y Sistemas, y 3º Workshop sobre Aspectos Teóricos de la Inteligencia Artificial, 7 al 9 de Noviembre de 1996, Universidad Nacional de San Luis, pp. 507-519.