CAPÍTULO 5. SOFTWARE

5.1. INTRODUCCIÓN

En este apartado se van a describir las aplicaciones software que se han creado expresamente para este proyecto. Dichas aplicaciones están programadas en el entorno de programación de objetos HP-VEE y son el desarrollo real de este Proyecto. Es lo esencial y lo que ha llevado más tiempo de ejecución.

La elección de este entorno es debida a:

- La experiencia que posee el grupo de investigación Cryobiotech (dirigido por D. Ramón Risco Delgado) en esta herramienta, muy utilizada en distintos Proyectos Fin de Carrera realizados en el departamento al que pertenece el proyecto que nos ocupa. Esta experiencia facilita la rápida resolución de posibles dudas que puedan surgir en el desarrollo de los programas.
- ✤ La facilidad de su uso.
- La empresa que ha proporcionado la PCI-DAS08 proporciona una librería muy fácil de instalar con la cual, el control de la tarjeta de adquisición de datos es muy asequible.
- El entorno gráfico que posee hace que el programa sea muy intuitivo y cómodo de usar para los médicos que, al final, van a ser los clientes finales que van a manejarlo.

Seguidamente se procederá a dar unas explicaciones breves del HP VEE para que se tenga una pequeña idea del mismo.

5.2. <u>ENTORNO DE PROGRAMACIÓN ORIENTADO A OBJETOS</u> <u>HP VEE</u>

A lo largo de la carrera se ha hecho más hincapié en lenguajes de programación secuenciales, muy importantes en cualquier aplicación hoy en día, en los cuales, se seguía una secuencialidad en la ejecución de las instrucciones, pero cada vez más se están usando lenguajes de programación orientado a objeto.

En estos lenguajes ya no se sigue una secuencialidad como en los anteriores, sino que dependiendo de que se produzcan una serie de eventos sobre unos objetos previamente definidos, se ejecutan unos fragmentos de códigos u otros.

Un objeto no es otra cosa que un trozo de código representado por una caja en HP VEE, de forma que, representando una funcionalidad, enmascara el código que tiene interiormente.

Por ejemplo, este objeto toma su entrada 'A' y realiza una operación aritmética con ella. El resultado de esa aplicación lo lleva a la salida por la variable 'Result'.



Fig. 5.2-1: Objeto 'Formula'

Evidentemente, esa entrada 'A' está conectada a otro objeto del cual es su salida y lo mismo sucede con 'Result'.

En este tipo de programación, son muy importantes las conexiones entre los distintos objetos, ya que ellas indican qué objetos se ejecutan primero y cuáles después. Con un ejemplo, todo lo que se quiere indicar se ve mucho mejor.



Fig. 5.2-2: Ejemplo 1 de HP VEE [8]

Este ejemplo tan sencillo crea una señal coseno y la representa.



Fig. 5.2-3: Ejemplo 2 de HP VEE

El programa anterior, cuando se pulsa Start, entra en un ciclo que se ejecuta continuamente. Crea un número aleatorio entre 0 y 1 y lo guarda en un registro que recuerda el número que se acaba de crear y el anterior. Los dos números que almacena el registro se comparan y, si el número previo es menor que el número actual, aparece un mensaje en pantalla diciendo 'Señal creciente'. En caso contrario, aparece el mensaje 'Señal decreciente'.

Este ejemplo representa un programa extremadamente sencillo, pero siguiendo la misma filosofía, se pueden crear programas de una gran complejidad.

A la hora de la implementación se han encontrado dificultades que ha sido necesario solventar. Una de ellas ha sido que, para que se active un objeto con varias entradas es necesario que se activen todas ellas en la misma iteración, ya que si no, dicho objeto no se ejecuta. Como se podrá comprobar a lo largo de la memoria que nos ocupa, este problema se ha presentado en muchas ocasiones. La forma de solventarlo ha sido colocando a la entrada unos objetos denominados 'Constant' (los hay de varios tipos, 'Integer', 'Real'... en función de las necesidades) que activaban ese valor cuando sea necesario en cada caso.

Las conexiones representan dos cosas: tanto datos (ya sean de entrada o de salida) como el orden que se ha de seguir en la activación de los objetos.

Por lo que se puede observar y como se dijo arriba, un objeto no es más que una parte del código con variables para realizar los cálculos necesarios.

Este tipo de programas vienen muy unidos a entornos gráficos, como es el caso del programa que nos ocupa, HP VEE. Gracias a esto, su programación se suele percibir bastante rápido, porque tienen entornos muy intuitivos y fáciles de captar para el programador.

Además, la modificación de un programa ya hecho es bastante simple. Consiste en añadir más objetos con sus respectivas conexiones.

Por último indicar que el programa tiene una ayuda muy potente capaz de aclarar una gran cantidad de problemas que surjan durante la realización del programa, y que existen unos manuales en formato PDF que vienen francamente bien cuando uno quiere empezar a programar en este lenguaje. Dichos ficheros son VEE_Pro_Users_Guide y VEE_Pro_Advanced_Techniques.

5.2.1. ENTORNO HP-VEE

Como se ha indicado anteriormente, HP-VEE es un lenguaje de programación orientado a objetos. Su entorno es el típico de cualquier aplicación que corre bajo Windows.



Fig. 5.2-4: Página inicial del entorno HP-VEE

Se pueden observar los distintos elementos de los que consta el entorno:

- <u>Barra de título</u>: Donde se pueden observar el nombre del programa y los botones típicos para Maximizar, Minimizar...
- Menú: En él se encuentran todas las funciones que se pueden usar y todos los objetos que se necesitan incluir en la creación de las aplicaciones. Es de destacar que para la utilización de la tarjeta de adquisición de datos PCI-DAS08 es necesario instalar la Universal Library for HP-VEE, la cual incluye un menú adicional en esta barra con el título 'DataAcq'.



Fig. 5.2-5: Menú 'DataAcq'

- <u>Barra de herramientas</u>: se tienen a nuestra disposición y de forma muy cómoda las funciones más necesarias para la creación de los programas.
- <u>Explorador de programa</u>: Se encuentra a la izquierda de la pantalla como se puede observar en la figura 5.2-4 y muestra la estructura jerárquica del programa. Resulta de mucha ayuda cuando la aplicación es extremadamente compleja.
- <u>Espacio de trabajo</u>: en él se colocan los objetos con sus respectivas uniones para crear el programa.
- <u>Barra de estado</u>: Muestra el estado de ejecución del programa.

Como es lógico pensar, no se pretende explicar la funcionalidad de cada uno de los objetos que están definidos ni de cada una de las funciones que se pueden usar ya que sería exageradamente extenso. Si se desea más información sobre cualquier tema en concreto, se puede acceder a los manuales antes mencionados VEE_Pro_Users_Guide y VEE_Pro_Advanced_Techniques.

Teniendo todo esto en cuenta, se procederá a explicar la aplicación.