

# Características técnicas del software de la Computadora Industrial Abierta Argentina

## 1. Introducción

En el presente documento se describen las características del software que poseerá la Computadora Industrial Abierta Argentina (CIAA). El software es el programa que se ejecuta en una computadora (Windows/Linux compatible) y permite: escribir el código que se ejecutará en la plataforma, compilar dicho código para la arquitectura del microcontrolador de la plataforma, descargar el programa al microcontrolador y visualizar/editar el mapa de memoria de la CIAA.

## 2. Equipo técnico a cargo del desarrollo de firmware de la CIAA

### Responsables:

- Responsable: Ing. J. Ezequiel Espósito (Debtech SRL, LSE-FIUBA)
- Subresponsable: Ing. Gustavo Alessandrini (INTI, ORT)

### Colaboradores:

- Ing. Paola Pezoimburu (LSE-FIUBA, Sur Emp. Tec.)
- Ing. Gustavo Alessandrini (INTI, ORT)
- Ing. Gustavo Muro (FCEIyA-UNR)

## 3. Documentos aplicables y de referencia

- Propuesta para el desarrollo de una Computadora Industrial Abierta Argentina:  
<http://www.sase.com.ar/asociacion-civil-sistemas-embebidos/files/2013/11/Propuesta-Computadora-Industrial-Abierta-Argentina-ACSE-CADIEEL-v2.5.pdf>
- Características técnicas del hardware de la Computadora Industrial Abierta Argentina:  
<http://www.sase.com.ar/asociacion-civil-sistemas-embebidos/files/2013/11/CIAA-Hardware-v1.1.pdf>
- Características técnicas del firmware de la Computadora Industrial Abierta Argentina:  
<http://www.sase.com.ar/asociacion-civil-sistemas-embebidos/files/2013/12/CIAA-Firmware-v1.0.pdf>

## 4. Requerimientos del software

Los requerimientos del software surgieron a partir de discusiones con los siguientes grupos de trabajo:

- CIAA-Hardware: grupo de trabajo encargado del hardware de la CIAA.
- CIAA-Firmware: grupo de trabajo encargado del firmware de la CIAA.
- CIAA-Software: grupo de trabajo encargado del software de la CIAA.
- CIAA-Coordinación: grupo de trabajo encargado del sistema CIAA a nivel general.

- Embebidos32@: comunidad abierta regional dedica a los sistemas embebidos.

Estos grupos de trabajo están formados principalmente por ingenieros provenientes del área industrial y del área académica, con amplia experiencia en el desarrollo y en la implementación de hardware, firmware y software para sistemas embebidos. A partir de la experiencia de los RRHH y de los conocidos estándares industriales se llegaron a los siguientes requerimientos de alto nivel para el software:

- Codificación de la CIAA en Lógica Ladder.
  - Mediante un editor gráfico.
  - Mediante un editor de texto.
- Codificación de la CIAA en C.
  - Mediante un editor de texto.
- Compilación del programa de la CIAA a un lenguaje intermedio que será interpretado por el firmware.
- Programación de de la CIAA.
  - A través de RS-485 ó Ethernet (descarga del programa al firmware).
- Comunicación con la CIAA.
  - Utilizando el protocolo estándar abierto Modbus.
  - A través de Ethernet.
  - A través de RS-232.
  - A través de RS-485.
- Mapa de memoria de la CIAA.
  - Edición del mapa de memoria.
  - Visualización del mapa de memoria.
- Interfaz Gráfica de Usuario.
  - Apertura y guardado de proyectos.
  - Interfaz gráfica de codificación.
  - Interfaz gráfica de compilación.
  - Interfaz gráfica de programación.
  - Interfaz gráfica del mapa de memoria.

## 5. Presentación conceptual del software

### 5.1. Principios de arquitectura de sistema

Para realizar el desarrollo del software se utilizarán los estándares más modernos aplicables a arquitectura de sistemas:

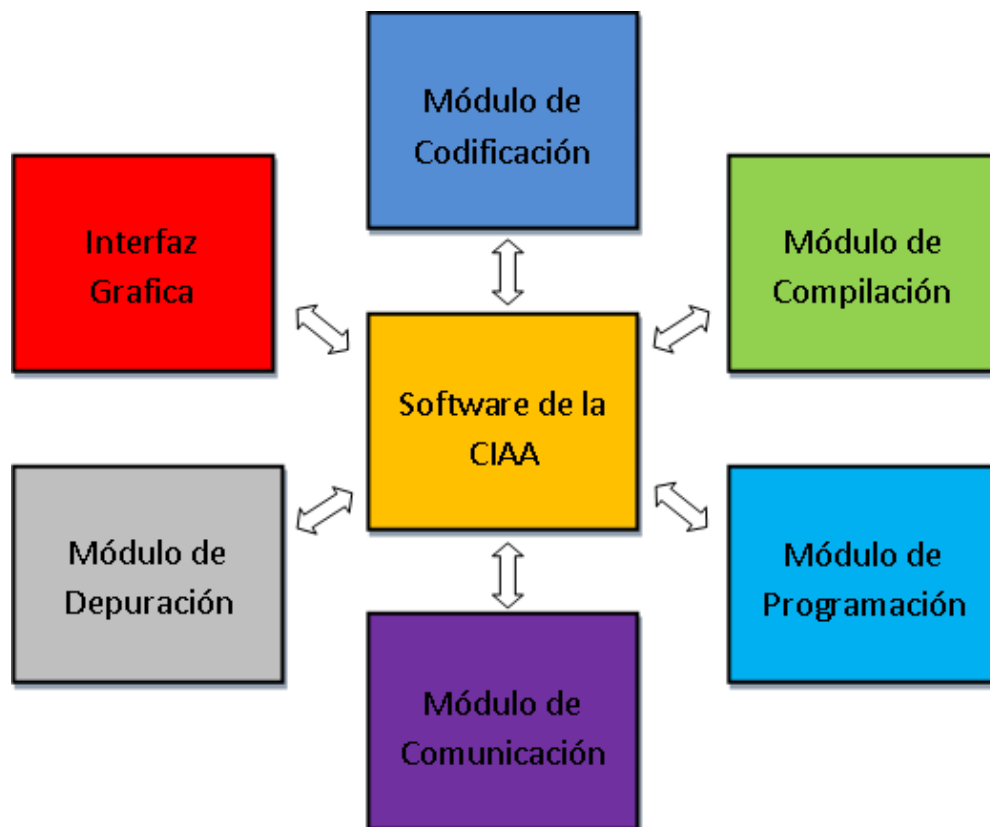
- Utilizar tecnologías Open-Source y Cross-Platform. La ventaja es que las aplicaciones se desarrollan una única vez y luego las mismas se podrán compilar para diferentes arquitecturas y sistemas operativos.
- Separar la capa de modelo de datos de la capa de interfaz gráfica. Para esto se utilizará el patrón de diseño MVC (Model View Controller). Las ventajas de utilizar este patrón de diseño, es que las aplicaciones resultan ser independiente de la biblioteca gráfica utilizada.
- Implementar protocolos de comunicación de manera tal que los mismos sean independientes del canal de transmisión. La principal ventaja de este esquema, es que los protocolos de comunicación industriales estándares se programarán una única vez y luego se los podrá utilizar en diferentes

medios (por ejemplo RS-233, RS-485 y Ethernet).

- Utilizar técnicas avanzadas de programación orientada a objetos, para escribir código flexible, escalable y mantenible.
- Utilizar el esquema servidor/cliente, comunicando a los mismos mediante servicios web RESTful y protocolo HTTP. Las principales ventajas de este esquema son, por un lado que el servidor (lógica de datos) es programado una única vez y los clientes (manejo y presentación de datos) pueden ser programados en diferentes lenguajes de programación ya que se utilizan interfaces RESTful, y por otro lado, que la comunicación entre el servidor y los clientes es transparente a las complicadas topologías de redes de las organizaciones ya que se utiliza el protocolo HTTP. Cabe aclarar, que por cuestiones de tiempo de desarrollo, en la primera versión del software de la CIAA se programará en conjunto el servidor con el cliente. En futuras versiones se pretende separar el servidor del cliente para generar un sistema más flexible, escalable y robusto.

## 5.2. Esquema general

Los módulos que comprenderán al software de la CIAA son:



### 5.3. Detalle de los módulos

**Módulo de Codificación:** este módulo es el que permite escribir los programas que luego ejecuta la CIAA. Los programas podrán ser escritos, por un lado, en Lógica de Ladder de manera gráfica y de mediante editor de texto. Por otro lado, también se podrán escribir programan en Lenguaje C mediante editor de texto.

**Módulo de Compilación:** este módulo es el encargado de compilar el programa escrito mediante el **Módulo de Codificación**. La compilación se hará a un lenguaje intermedio que luego será interpretado por el firmware.

**Módulo de Programación:** este módulo es el encargado de tomar el programa compilado en un lenguaje intermedio (que es generado por el **Módulo de Compilación**) y enviarlo al microcontrolador de la CIAA. Para esto se utilizará la interfaz RS-485 o Ethernet. El programa en lenguaje intermedio será almacenado en las memoria disponibles en la CIAA. Es decir los pasos para generar un programa que se ejecute en la CIAA serán:

1. Códificación en Lógica Ladder o en C utilizando el **Módulo de Codificación**.
2. Compilación del programa a un lenguaje intermedio (por ejemplo un formato XML).
3. Programación de la CIAA a partir de enviar el programa en lenguaje intermedio al firmware (mediante RS-485 o Ethernet).
4. El firmware se encargará de guardar el programa en la memoria SD disponibles por hardware, realizado las comprobaciones de integridad necesarias y tomando las precauciones de robustez adecuadas.
5. El firmware enviará el programa guardado en la SD a la flash interna del correspondiente microcontrolador.
6. El firmware interpretará el programa guardado en la flash y lo ejecutará paso a paso.

**Módulo de Comunicación:** este módulo será el encargado de establecer comunicaciones con la CIAA. Dicha comunicación se establecerá mediante el protocolo industrial estándar Modbus utilizando Ethernet, RS-232 o RS-485.

**Módulo de Depuración:** este módulo será el encargado de obtener información del mapa de memoria de la CIAA, para su visualización y/o edición.

**Interfaz Gráfica:** es la interfaz gráfica para el usuario. La misma le permitirá a los usuarios de la plataforma CIAA realizar la codificación, compilación, programación y depuración de manera gráfica.

## 6. Selección de tecnología de desarrollo

A través de los grupos de trabajo anteriormente mencionados, se sometió a debate cuál debería ser las tecnologías utilizadas para desarrollar el software de la CIAA. Las opciones fueron:

Lenguaje de Programación	Biblioteca Gráfica	Bibliotecas	Herramientas Libres
C++	Qt	Boost	Binutils, gcc, newlib y OpenOCD
Java	Framework Eclipse	Boost	Binutils, gcc, newlib y OpenOCD
Java	Framework JavaFX	Boost	Binutils, gcc, newlib y OpenOCD

Por disponer de poco tiempo para realizar el desarrollo y por los conocimientos generales que maneja la comunidad a nivel regional, la opción Java | Framework Eclipse | Boost | Herramientas Libres quedó descartada. La justificación se basa en el hecho de que la comunidad de desarrollo a nivel regional no tiene amplios conocimientos y experiencia en el Framework Eclipse. Si se utilizará este conjunto de tecnologías el desarrollo tomaría mucho más tiempo del planeado (aproximadamente de 6 a 8 meses desde la publicación de los documentos) y no se estarían cumpliendo con los objetivos del proyecto.

Por otro lado, por ser el Framework JavaFX una tecnología muy poco madura en el mercado, la opción Java | Framework JavaFX | Boost | Herramientas Libres quedó descartada. La justificación es que al ser la CIAA una plataforma industrial se requiere que tanto el hardware, como el firmware y como el software sean robustos. Al ser JavaFX una tecnología muy nueva, la condición de robustez no estaría siendo cumplida.

Finalmente, se decide que las tecnologías más apropiadas para llevar a cabo el desarrollo son C++ | Qt | Boost | Herramientas libres. La justificación es que por un lado a nivel regional, la comunidad tiene gran conocimiento y experiencia en estas tecnologías. Por otro lado, la cantidad de años de presencia que tienen estas tecnologías en el mercado y la cantidad de proyectos exitosos que se desarrollaron a nivel internacional y nacional, hacen que estas herramientas sean las correctas en términos de confianza, y robustez.

## 7. Herramientas de desarrollo

Las herramientas de desarrollo a utilizar son:

- Administración:
  - Wiki para el desarrollo del proyecto: <http://www.proyecto-ciaa.com.ar/devwiki>
  - Administración del proyecto y bug tracker: <http://www.proyecto-ciaa.com.ar/project>
- Desarrollo:
  - IDE de desarrollo: <http://qt-project.org/wiki/Category:Tools::QtCreator>
- Bibliotecas:
  - Biblioteca gráfica
  - Boost: <http://www.boost.org/>

## 8. Lista de tareas del software

En esta sección se definen las listas de tareas a realizar para cada módulo, el encargado de llevar adelante cada tarea y el tiempo de desarrollo de las mismas.

### 8.1. Lista de tareas para el Módulo de Codificación

N°	N° Prev	Detalle de la Tarea	Encargado	Horas Hombre
1	-	Implementación de las clases que permitan definir los elementos de la de Lógica Ladder		
2	1	Implementación de las clases que permitan combinar elementos de Lógica Ladder para generar un programa		
3	2	Generación de programa de Lógica Ladder mediante editor gráfico para la CIAA		
4	2	Generación de programa de Lógica Ladder mediante editor de texto para la CIAA		
6	-	Generación de programa en Lenguaje C mediante editor de texto para la CIAA		

### 8.2. Lista de tareas para el Módulo de Compilación

N°	N° Prev	Detalle de la Tarea	Encargado	Horas Hombre
1	-	Traductor del programa en Lógica Ladder en un lenguaje intermedio que será interpretado por el firmware de la CIAA		
2	-	Traductor del programa en Lenguaje C en un lenguaje intermedio que será interpretado por el firmware de la CIAA		

### 8.3. Lista de tareas para el Módulo de Programación

N°	N° Prev	Detalle de la Tarea	Encargado	Horas Hombre
1	-	Envío al firmware de la CIAA el programa en lenguaje intermedio mediante interfaz RS-485		
2	-	Envío al firmware de la CIAA e lprograma		

		en lenguaje intermedio mediante interfaz Ethernet		
--	--	---	--	--

#### 8.4. Lista de tareas para el Módulo de Comunicación

N°	N° Prev	Detalle de la Tarea	Encargado	Horas Hombre
1	-	Implementación de la interfaz de comunicación genérica (apertura de puerto, escritura, lectura y cerrado de puerto)		
2	1	Implementación de las clases que permitan manejar el Puerto Serie de la PC		
3	1	Implementación de las clases que permitan manejar Sockets Ethernet desde la PC		
4	2,3	Implementación del protocolo estándar Modbus		

#### 8.5. Lista de tareas para el Módulo de Depuración

N°	N° Prev	Detalle de la Tarea	Encargado	Horas Hombre
1	-	Obtención de variables del mapa de memoria de la CIAA para visualización a partir de la comunicación Modbus		
2	1	Edición de variables del mapa de memoria de la CIAA a partir de la comunicación Modbus		

#### 8.6. Lista de tareas para la Interfaz Gráfica

N°	N° Prev	Detalle de la Tarea	Encargado	Horas Hombre
1	-	Gestión de proyectos. Creación,		

		guardado, apertura y cerrado de proyectos.		
2	-	Interfaz gráfica para codificación.		
3	-	Interfaz gráfica para compilación.		
4	-	Interfaz gráfica para programación.		
5	-	Interfaz gráfica para depuración.		

### 8.7. Lista de tareas para la Validación del Software

N°	N° Prev	Detalle de la Tarea	Encargado	Horas Hombre
1	-	Validación del Módulo de Codificación.		
2	-	Validación del Módulo de Compilación.		
3	-	Validación del Módulo de Programación.		
4	-	Validación del Módulo de Depuración.		
5	-	Validación de la Interfaz Gráfica.		

### 9. Requisitos de documentación para los desarrolladores del software

- Bug Tracker:
  - <http://www.proyecto-ciaa.com.ar/project>
- Guía de estilo y convención de nombres:
  - <http://google-styleguide.googlecode.com/svn/trunk/cppguide.xml>
- Documentación de archivos de código fuente para soportar formato Doxygen
  - [www.doxygen.org](http://www.doxygen.org)