



QuantumLab
Sistema de Simulación y Visualización de
Operaciones de Control en Qubits

Desarrollado por:

Jesús Fuentes Aguilar

Índice general

1. Arquitectura	1
1.1. Acerca de QuantumLab	1
1.1.1. Módulos	2
1.1.2. Archivos	4
1.2. Requerimientos No Funcionales	6
1.2.1. Hardware	6
1.2.2. Software	6
2. Manual de Usuario	7
2.1. Instalando QuantumLab	7
2.2. Ejecutar QuantumLab	8
2.3. Trabajando con QuantumLab	9
2.3.1. Ventana de parámetros de simulación	9
2.3.2. Ventana de la esfera de Bloch	9
2.3.3. Ventana de análisis de datos	11
2.3.4. Ventana de coordenadas	11

ÍNDICE GENERAL

1

Arquitectura

1.1. Acerca de QuantumLab

QuantumLab es un simulador de sistemas cuánticos de dos niveles accesibles para evaluar protocolos de control cuántico que dictaminen las secuencias que pueden ser implementadas físicamente para accionarlas como compuertas lógicas.

Se trata de un sistema que incorpora aspectos visuales y gráficos en la presentación de los resultados, para lo cual se utiliza la esfera de Bloch; además, genera un análisis sobre la evolución temporal del sistema como apoyo al procedimiento de simulación.

Este sistema está dirigido a los físicos dentro del área del cómputo cuántico que requieran evaluar modelos propuestos de qubits para determinar su comportamiento y contribuir a la presentación de resultados fieles a través de un sistema que permita una flexible manipulación gráfica de los resultados.

La parte matemática y de cómputo de QuantumLab se desarrolla en lenguaje C++, mientras que la parte gráfica queda basada en el Ambiente de Desarrollo de Interfaz Gráfica de Usuario de Matlab. A fin de que los componentes escritos en C++ se enlacen con aquellos componentes escritos en Matlab, de manera transparente al usuario final, se recurre a la Interfaz de Programación de Aplicaciones de Matlab.

1. ARQUITECTURA

1.1.1. Módulos

En esta fase, de desarrollo y evaluación, el proyecto se orienta únicamente a sistemas UNIX. La simulación se ha escrito en lenguaje C++ y se apoya, solamente, del motor de Matlab para ofrecer al usuario una interfaz gráfica que despliegue los resultados e indique qué parámetros ingresar para accionar al sistema. En concreto, el usuario debe ser ajeno a los procedimientos que el sistema de visualización tenga que hacer para presentarle los resultados. Para ello se diseña la estructura interna del sistema de visualización relacionando entre sí a tres unidades principales:

Simulador en C++ Esta unidad es la que procesa toda la información que recibe del usuario para procesarla de acuerdo con las ecuaciones desarrolladas en capítulos anteriores y devolver una serie de resultados que serán, más tarde, leídos de un archivo de datos.

Interfaz Gráfica de Usuario Se construye en Matlab. Sus funciones son las de capturar los parámetros de simulación (por parte del usuario), desplegar los resultados de manera gráfica y ofrecer una serie de herramientas para el análisis de estados cuánticos sobre la esfera de Bloch.

Administrador del motor de Matlab en C++ Esta unidad enlaza al Simulador con la Interfaz Gráfica de Usuario mediante un proceso transparente al usuario. Indica al sistema operativo que de apertura (y cierre) al motor de Matlab, e integra todos los procedimientos para comunicar a las otras dos unidades. A través de esta aplicación se inicia el simulador **QuantumLab**.

Para unir los procesos de simulación con los de visualización, se diseña una interfaz gráfica de usuario, a través de la cual el usuario tiene el control para introducir los parámetros de simulación (o de entrada), visualizar los resultados y aprovechar las herramientas de análisis y las de manipulación gráfica.

El diagrama de casos de uso en la figura 1.1 muestra todas aquellas funcionalidades que el sistema (**QuantumLab**) ofrece al usuario final. Como se puede ver, es a través del Administrador del motor de Matlab en C++ desde el cual el usuario final da inicio al sistema, dándole acceso a la Interfaz Gráfica de Usuario mediante sus dos ventanas principales: la ventana de parámetros de simulación y la ventana de la esfera de Bloch. Así el sistema es presentado como un único proceso ante el usuario final, es decir, todos aquellos procesos entre la unidad de simulación en C++ con las unidades gráficas quedan ocultos al usuario.

1.1 Acerca de QuantumLab

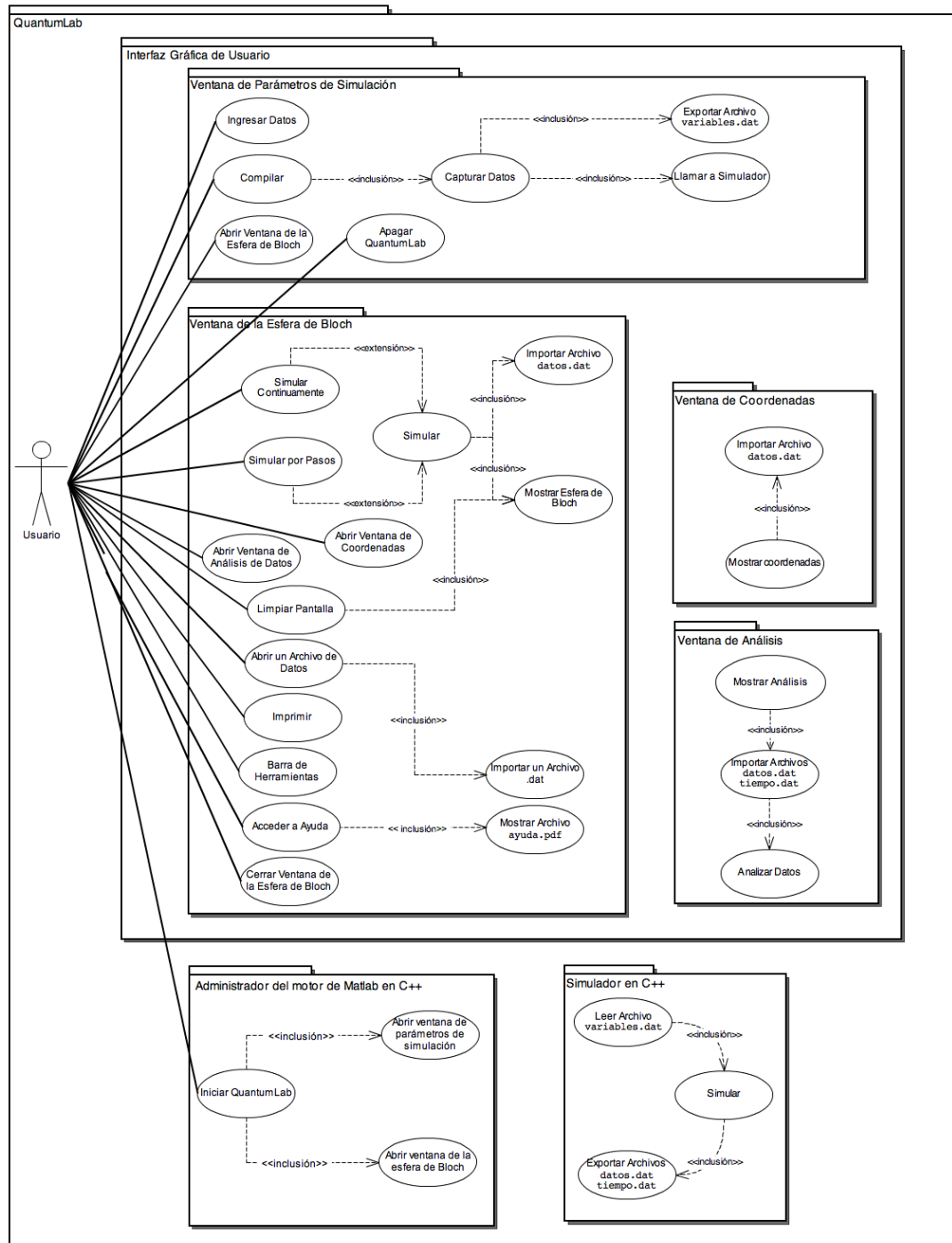


Figura 1.1: Diagrama de Casos de Uso - Se muestran las funcionalidades del sistema entero: **QuantumLab**. Su implementación se describe en el siguiente capítulo.

1. ARQUITECTURA

La interfaz gráfica de usuario se compone de las siguientes unidades:

Ventana de Parámetros de Simulación Permite ingresar los datos de entrada para efectuar la simulación.

Ventana de la Esfera de Bloch Muestra los resultados de la simulación sobre la S^2 esfera.

Ventana de Análisis de Datos Ejecuta y muestra el análisis de los resultados de la simulación, como tiempo de simulación, la coordenada en z máxima y en z mínima, y el número de veces que la evolución temporal del sistema cruza por el ecuador.

Ventana de Coordenadas Muestra las coordenadas correspondientes a un estado del sistema en función del ciclo del pulso del campo armónico externo que se aplica sobre el qubit.

1.1.2. Archivos

El disco contiene el código fuente de cada uno de los componentes que conforman **QuantumLab**. También se incluyen algunos otros archivos que han formado parte del desarrollo del simulador y, otros más, que se anexan para su consulta. A continuación se describe el contenido del disco.

Directorio raíz /

QuantumLab/ Carpeta que contiene todos los componentes que integran **QuantumLab** y misma que debe ser copiada al equipo donde se desee ejecutar este simulador.

Carpeta **QuantumLab/**

Simulador 1.0/ Carpeta con las carpetas y los archivos de código fuente correspondientes a la primera fase de desarrollo del simulador.

Simulador 1.1/ Carpeta con los archivos de código fuente correspondientes a la fase corriente de desarrollo del simulador.

ayuda.pdf Archivo que auxilia al usuario final en los pasos para la ejecución del sistema.

Analisis.m Código fuente del componente de la interfaz gráfica de usuario que muestra el análisis de datos de la simulación.

1.1 Acerca de QuantumLab

Coordenadas.m Código fuente del componente de la interfaz gráfica de usuario que muestra las coordenadas y el ciclo asociados a la evolución de estados cuánticos.

EsferaBloch.m Código fuente del componente de la interfaz gráfica de usuario que muestra la esfera de Bloch y despliega los resultados sobre su superficie.

QuantumLab.m Código fuente del componente de la interfaz gráfica de usuario que permite al usuario final ingresar los parámetros de simulación.

Analisis.fig Archivo soporte en el procesamiento de los gráficos utilizados por el componente **Analisis.m**

Coordenadas.fig Archivo soporte en el procesamiento de los gráficos utilizados por el componente **Coordenadas.m**

EsferaBloch.fig Archivo soporte en el procesamiento de los gráficos utilizados por el componente **EsferaBloch.m**

QuantumLab.fig Archivo soporte en el procesamiento de los gráficos utilizados por el componente **QuantumLab.m**

inicio Archivo ejecutable para iniciar **QuantumLab**

qubit Archivo ejecutable para el proceso de simulación invocado por el archivo **QuantumLab.m**.

Carpeta **QuantumLab/Simulador 1.0/**

Simulador 1.0x/ Carpeta con el código archivos de código fuente de una versión del simulador en Matlab.

bloch.m Rutina en Matlab para graficar el contenido de un archivo de datos provisto por **simsistema.cpp**

simfunonda.cpp Primera fase del simulador: archivo de código fuente para calcular la función de onda.

simsistema.cpp Primera fase del simulador: archivo de código fuente para mapear la función de onda y obtener las coordenadas sobre la esfera de Bloch. Este archivo exporta los datos que al ejecutar **bloch.m** en Matlab son graficados sobre la esfera de Bloch.

Carpeta **QuantumLab/Simulador 1.0/Simulador 1.0x/**

1. ARQUITECTURA

`qcampo.m` Rutina en Matlab para simular un sistema cuántico de dos niveles en presencia de un campo armónico externo.

`qlibre.m` Rutina en Matlab para simular un sistema cuántico de dos niveles sin la presencia de un campo armónico externo.

Carpeta `QuantumLab/Simulador 1.1/`

`inicio.cpp` Archivo de código fuente para mandar llamar al motor de Matlab e iniciar una sesión de **QuantumLab**.

`qubit.cpp` Archivo de código fuente de la fase corriente de desarrollo del simulador. Se puede decir que es el corazón del sistema.

`ciclos.h` Inicialmente fue creada para definir el número de ciclos que admite el sistema, por alguna razón que luego se despreció. Conforme el desarrollo del sistema avanzó, el archivo de cabecera se dejó para que el número máximo de ciclos que admite el sistema pueda ser configurado, sin alterar el código en C++.

1.2. Requerimientos No Funcionales

La estabilidad del simulador QuantumLab se ha probado siguiendo las siguientes características técnicas:

1.2.1. Hardware

Probado sobre:

Procesador	Intel Core 2 Duo 2.26 GHz
Memoria	4 GB 1067 MHz DDR3
Gráficos	NVIDIA GeForce 9400M 256 MB

1.2.2. Software

Probado sobre:

Sistema Operativo	Mac OSX 10.7.2
Versión de Matlab	Matlab 7.12.0
Versión de GCC/G++	4.2.1 (Based on Apple Inc. build 5658)

2

Manual de Usuario

2.1. Instalando QuantumLab

1. En los sistemas UNIX, hay que declarar algunas variables de entorno para dirigir las librerías dinámicas a las carpetas donde se encuentran los archivos de configuración y opciones del motor de Matlab. Este procedimiento es necesario para compilar un código en C++ cualesquiera que haga uso de las librerías del motor de Matlab, de otro modo, la compilación no se puede efectuar.

Hay que dirigirse al archivo de configuraciones de perfil y declarar la ruta a la que se deben dirigir las librerías dinámicas, para ello abrir una terminal y en la línea de comandos escribir:

```
:~ usuario$ pico .bash_profile
```

Una vez abierto el archivo de configuraciones de perfil, agregar las líneas:

```
export DYLD_LIBRARY_PATH=$DYLD_LIBRARY_PATH:/Applications/  
<matlabversion>.app/bin/<arquitectura>:/Applications/  
<matlabversion>.app/sys/os/<arquitectura>  
  
export PATH=$PATH:/Applications/<matlabversion>.app/bin
```

Cuadro 2.1: Declaración de variables de entorno en UNIX.

Del cuadro 2.1, los valores <arquitectura> y <matlabversion>, corresponden a la arquitectura del sistema y al nombre bajo el cual se instaló la versión de Matlab en el equipo, respectivamente.

2. MANUAL DE USUARIO

2. Copiar la carpeta **QuantumLab/** del disco a algún directorio en el equipo.

NOTA: No modificar ni borrar ningún archivo del código fuente que compone a **QuantumLab**, de otra forma el simulador puede experimentar fallas e inestabilidad propiciando un mal desempeño.

2.2. Ejecutar QuantumLab

La rutina `inicio`, puede ser ejecutada desde la terminal de comandos dirigiéndose a la carpeta **QuantumLab/** y escribiendo:

```
:QuantumLab usuario$ ./inicio
```

Las ventanas principales del simulador se abren a continuación, ver figura 2.1. Durante la ejecución de **QuantumLab**, la ventana de comandos no debe ser cerrada, pues se produce inestabilidad en el sistema y se pierde el control para indicarle al sistema operativo cuándo finalizar el proceso en turno.

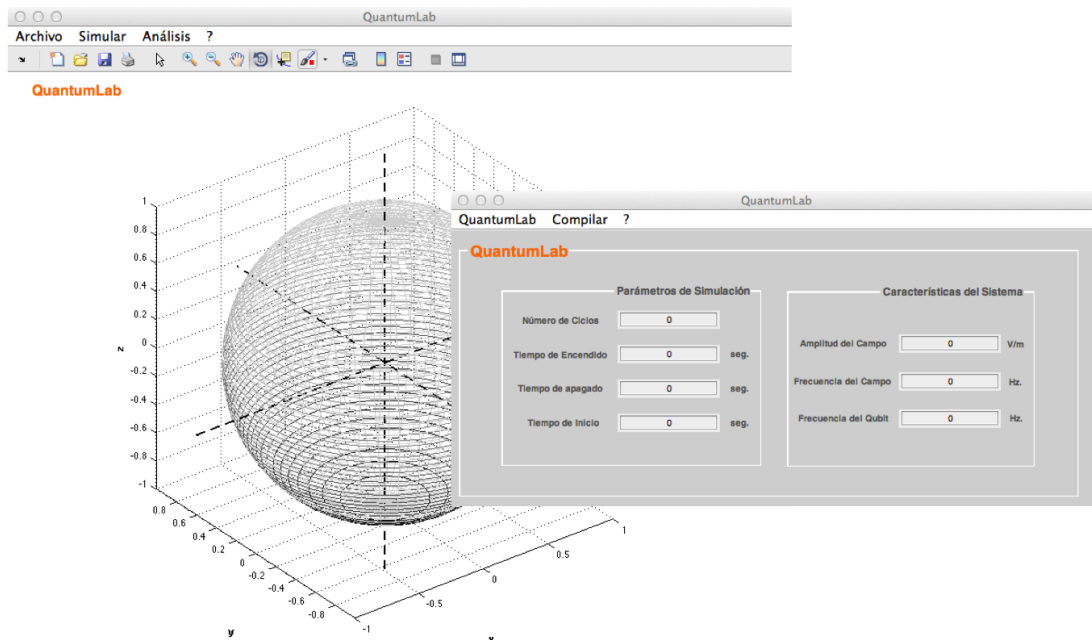


Figura 2.1: Ventanas principales de **QuantumLab** - Interfaz gráfica de usuario para empleo del simulador.

2.3. Trabajando con QuantumLab

QuantumLab ofrece un ambiente gráfico al usuario para ingresar los datos que caracterizan al sistema y para desplegar los resultados de la simulación. A través de esta interfaz el usuario puede analizar los resultados, ya sea gráficamente por medio de la esfera de Bloch, o bien, analizar las coordenadas de cada estado cuántico mediante un menú que muestra estos datos, o incluso ambas.

2.3.1. Ventana de parámetros de simulación

Es la unidad gráfica donde el usuario final ingresa los parámetros de entrada para evaluar el protocolo de control, figura 2.2. También a través de ésta, se compilan los datos que caracterizan al sistema llamando (internamente) al simulador. Una vez que se han compilado los parámetros de simulación, es posible analizarlos por las otras tres unidades que componen la interfaz gráfica de usuario.

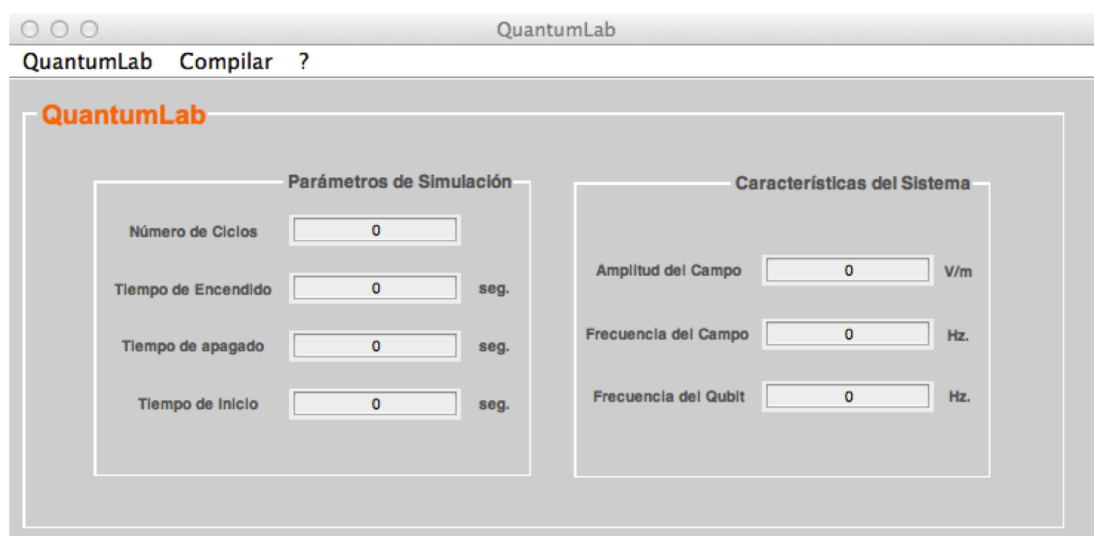


Figura 2.2: Interfaz gráfica de usuario: ventana de parámetros de simulación - El usuario final ingresa los parámetros de simulación en los campos indicados para luego compilarlos –proceso que toma lugar mediante la intervención del simulador en C++, el cual es llamado al hacer actuar la compilación.

2.3.2. Ventana de la esfera de Bloch

Esta unidad muestra la esfera de Bloch, en la cual se visualiza la evolución temporal del sistema en términos de coordenadas (x, y, z) sobre la superficie de ésta.

2. MANUAL DE USUARIO

A través de esta ventana, figura 2.3, se tiene acceso a las ventanas de coordenadas y a la ventana de análisis de datos.

Se puede acceder a dos tipos de simulación a través de esta ventana, una vez que ha compilado los datos de entrada en la ventana de parámetros de simulación.

La simulación ordinaria, grafica la evolución temporal del sistema de una sola vez, mientras que, la simulación por pasos grafica la evolución temporal del sistema de acuerdo al ciclo del pulso en turno. La ventaja de la simulación por pasos es que permite visualizar el estado del sistema uno por uno conforme suceda, siendo que la simulación ordinaria muestra *estáticamente* el comportamiento del sistema.

Una barra de herramientas permite manipular la esfera para apreciar los datos representados sobre ésta a decisión del usuario final.

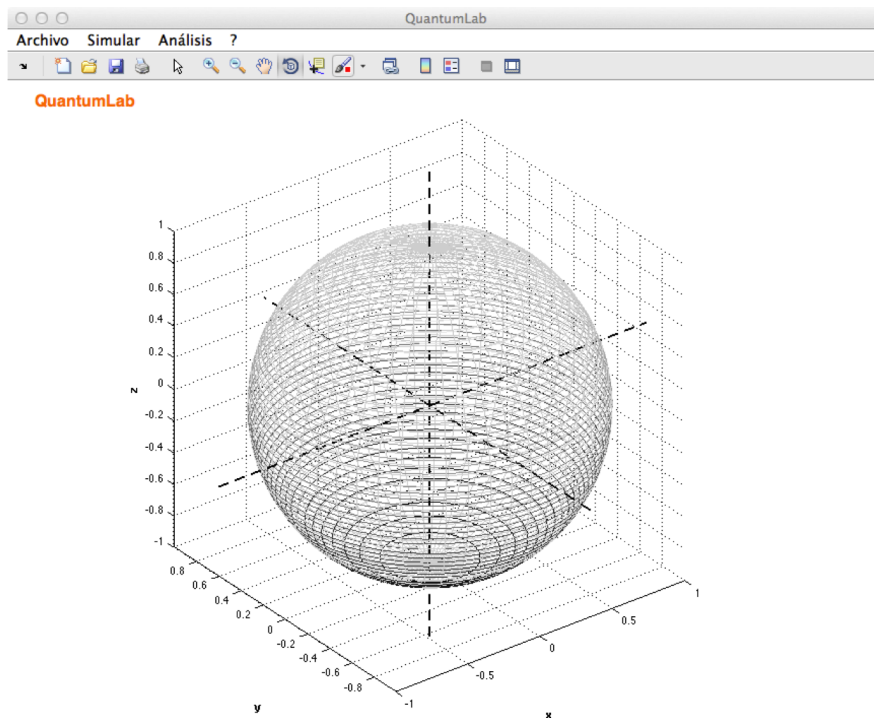


Figura 2.3: Interfaz gráfica de usuario: ventana de la esfera de Bloch - Esta ventana se muestra por omisión al haber iniciado QuantumLab, aunque también puede ser accedida desde la ventana de parámetros de simulación. Para accionar la simulación se tienen dos opciones: la simulación ordinaria y la simulación por pasos. Una barra de herramientas permite manipular la esfera de manera flexible.

2.3.3. Ventana de análisis de datos

Es una ventana que reúne algunos datos importantes a considerar sobre la evolución temporal del sistema, ver la figura 2.4. Se accede directamente desde la ventana de la esfera de Bloch y evalúa los datos una vez que el usuario final ha compilado los parámetros de entrada.

Para evaluar el desempeño de un protocolo de control se hace necesario conocer cuál es el estado más próximo para realizar una operación de compuerta y realizar ajustes para volver a evaluar el protocolo de control. La transición perfecta del estado $|0\rangle$ al estado $|1\rangle$ es, en coordenadas (x, y, z) , $(0, 0, 1) \rightarrow (0, 0, -1)$, y viceversa.

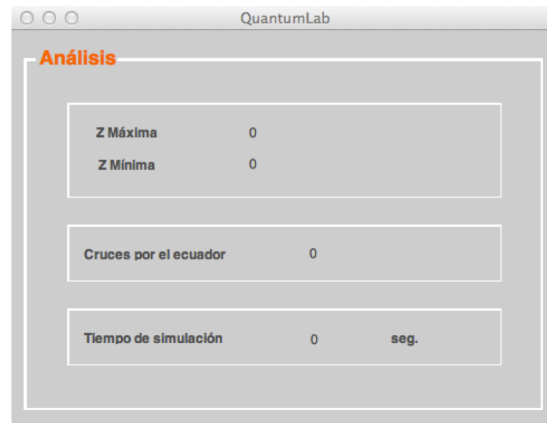


Figura 2.4: Interfaz gráfica de usuario: ventana de análisis de datos - Desde esta ventana se muestran datos relevantes para favorecer el análisis sobre el protocolo de control, por ejemplo la coordenada en z máxima y la coordenada en z mínima, el tiempo de simulación y las veces que el estado del sistema ha cruzado por el ecuador.

2.3.4. Ventana de coordenadas

Es una ventana sencilla que muestra las coordenadas (x, y, z) y el número de ciclo N correspondientes a cada estado cuántico, figura 2.5. Junto con la ventana de análisis, favorecen la evaluación del protocolo de control aplicado. Esta ventana se accede desde la ventana de la esfera de Bloch.

2. MANUAL DE USUARIO



Figura 2.5: Interfaz gráfica de usuario: ventana de coordenadas - Muestra las coordenadas (x, y, z) que se asocian a cada estado cuántico del sistema, además del correspondiente número de ciclo de pulso aplicado al qubit.