

Proyecto NSNyst Controller

Descripción del Proyecto y Requisitos

Versión 0.1

Ing. Roberto Antonio Becerra García

Contenidos

Descripción del Proyecto	3
Requisitos Funcionales	5
Estimulación	5
Visualización	6
Registro	6
Gestión de Estudios	7
Gestión de Protocolos	7
Gestión de Clasificaciones	8
Configuración	8
Requisitos Técnicos	10
Adquisición	10
Almacenamiento	10
Proceso de Estimulación-Captura-Visualización	11
Tareas Iniciales	12

Descripción del Proyecto

El NSNyst es un proyecto para el diseño y construcción de un electronistagmógrafo digital utilizando la técnica denominada electrooculografía. Este proyecto es fruto de la colaboración entre el Grupo de Procesamiento de Datos Biomédicos (GPDB) de la Universidad de Holguín (UHO) y el Departamento de Tecnología Electrónica (TE) de la Universidad de Málaga (UMA). El proyecto tiene como objetivo principal la creación de un equipo de bajo coste y portable que permita su utilización en aplicaciones clínicas e investigaciones en el ámbito de la medicina.

El componente de hardware del NSNyst por si mismo no es suficiente para su explotación por parte del personal al que está destinado. Para ello se hace necesario elaborar las aplicaciones de firmware y software que se encarguen de tareas como la estimulación visual, adquisición y visualización de las señales capturadas por el amplificador del NSNyst. Estas aplicaciones no existen en la actualidad por lo que el objetivo principal de este proyecto es su desarrollo.

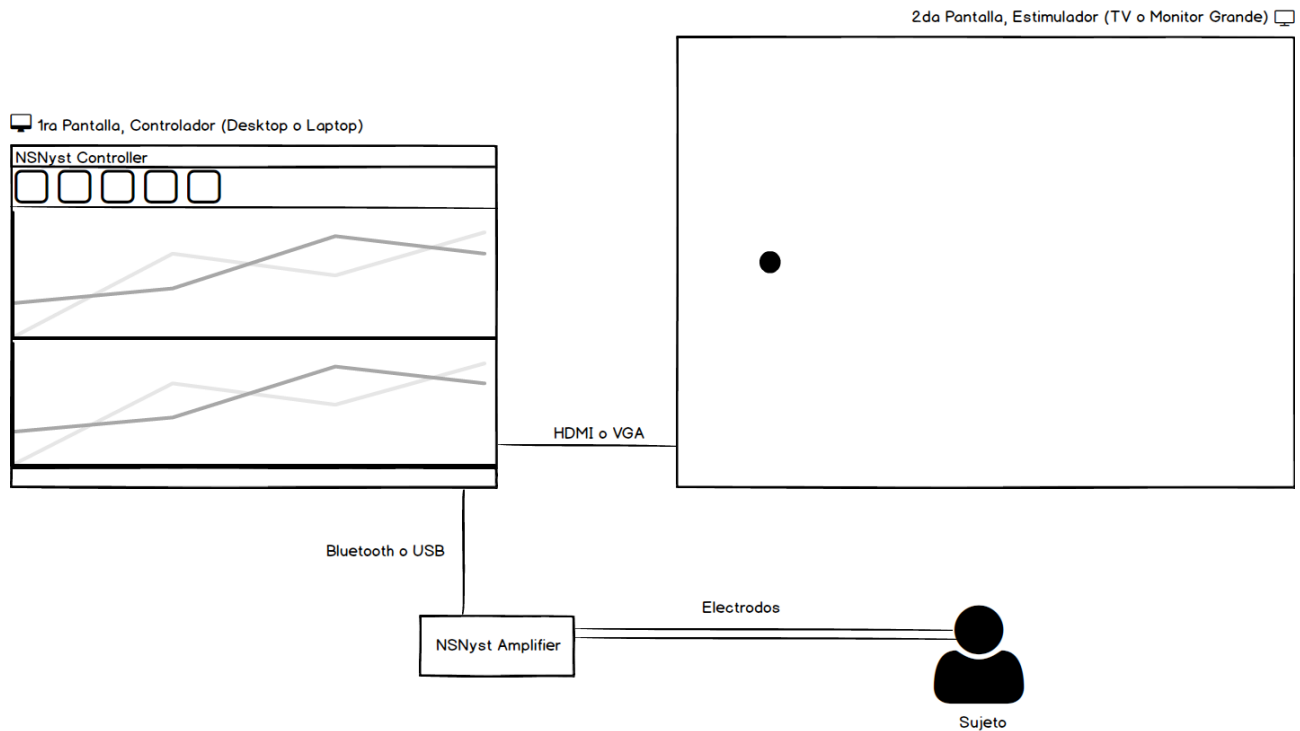
Para la tarea de adquisición se hace necesaria la implementación del firmware del microcontrolador encargado del envío de las señales digitales al ordenador. Además se necesita un protocolo de comunicación y su implementación del lado del ordenador, de forma que sea posible capturar las señales con una frecuencia de muestreo de 1 Khz en tiempo real. Se utilizará una interfaz serial ya sea USB o Bluetooth.

Para la utilización de este equipamiento en aplicaciones de carácter clínico es necesario la implementación de pruebas como las sacádicas o de persecución. Estas se definen básicamente por el estímulo visual seleccionado. Por esto se hace necesario incluir en la aplicación un estimulador capaz de generar los diferentes estímulos visuales utilizados en las pruebas y permitir un nivel de configuración adecuado. La salida del estimulador visual se mostrará en una pantalla secundaria conectada por la interfaz VGA o HDMI. Es importante que se tenga en cuenta que se utilizarán distintas pantallas de diferentes tamaños por lo que esta es una variable importante a tener en cuenta en la configuración en general de la aplicación.

Es importante la posibilidad de diseñar protocolos de pruebas que serán aplicados a conjuntos de sujetos para estudios experimentales. Es decir la aplicación deberá brindar posibilidades para el diseño y ejecución de pruebas. En el caso de la ejecución se deberán almacenar las señales adquiridas de cada una de las pruebas con los parámetros asociadas a las mismas y los datos básicos del sujeto registrado.

Para el personal médico especialista es de vital importancia la posibilidad de visualizar en tiempo real las señales adquiridas. Esto permitirá tomar decisiones como por ejemplo la de cancelar la ejecución de las pruebas por mal funcionamiento del equipo o por mala colocación de los electrodos.

En general el siguiente esquema muestra una configuración típica del NSNyst:



La pantalla principal (1ra pantalla) podrá ser tanto la de un ordenador de escritorio como uno portátil y es donde se mostrarán las señales que se adquieran y las funciones de control del equipamiento. A este ordenador se conectará el *NSNyst Amplifier* sea ya por USB o de forma inalámbrica por Bluetooth. La segunda pantalla será utilizada con el mecanismo de monitor extendido ofrecido por el sistema operativo y será conectada por HDMI o VGA.

Requisitos Funcionales

El proyecto cuenta con 7 unidades funcionales que se listan a continuación:

- **Estimulación:** Diseño, almacenamiento y ejecución de estímulos visuales.
- **Visualización:** Visualización de la captura y de señales previamente capturadas.
- **Registro:** Rutinas de control del registro.
- **Gestión de Estudios:** Todo lo relacionado con el manejo de estudios.
- **Gestión de Protocolos:** Todo lo relacionado con el manejo de protocolos.
- **Gestión de Clasificaciones:** Todo lo relacionado con el manejo de clasificadores de estudios.
- **Configuración:** Aspectos configurables de la aplicación.

En los siguientes epígrafes detallaremos todos los requisitos de cada uno de estos.

Estimulación

La unidad de estimulación es la encargada en general de diseñar y ejecutar los diferentes patrones de estimulación necesarios en las distintas pruebas médicas en las que se utilicen.

El primer requisito es el de *diseñar estímulo*, para esto la aplicación contará con dos tipos de estímulos: sacádicos y de persecución. En general todos los estímulos contarán con 3 valores configurables:

- Nombre: Cadena de texto con el nombre asignado al estímulo.
- Canales: Canales involucrados en el estímulo. Este valor podrá ser Horizontal, Vertical o Ambos.
- Duración: Tiempo en milisegundos que dura el estímulo completo.

Además de los estímulos sacádicos se podrá configurar los siguiente:

- Amplitud: Amplitud del movimiento en grados.
- Duración: Duración aproximada de la fijación en milisegundos.
- Variación: Variación del tiempo de la fijación en milisegundos.

Es importante señalar que la duración de cada fijación se calculará con la siguiente fórmula:

$$\text{VariaciónFinal} = \text{Duración} + \text{random}(\text{Variación} * 2) - \text{Variación}$$

De los estímulos de persecución se configurará lo siguiente:

- Amplitud: Amplitud del movimiento en grados.
- Velocidad: Velocidad del estímulo en grados por segundo.

Los parámetros de los estímulos que se diseñen deberán ser almacenados en formato JSON en un fichero local denominado algo así como **stimuli.json**.

El segundo requisito de esta unidad funcional es *ejecutar estímulo*. Este requisito consiste en la visualización del estímulo visual diseñado en la pantalla secundaria de la aplicación. Esto se debe ejecutar en modo de pantalla completa para que no hayan distracciones para el sujeto de prueba. Hay que tener en cuenta que hay que convertir las unidades físicas (grados) a píxeles en pantalla teniendo en cuenta aspectos como el tamaño de la pantalla y su resolución. Otro aspecto ha tener en cuenta es la tasa de refresco de la pantalla (típicamente 60 Hz) para optimizar el rendimiento del motor de estimulación y no dibujar más cuadros de los que se visualizan en la pantalla.

Es importante señalar que este motor debe ser lo más eficiente como sea posible debido a que las tareas de estimulación deben ejecutarse concurrentemente con la de captura y visualización. La generación gráfica del estímulo debe ser generada en un hilo aparte y solo será volcada por el hilo principal de la aplicación.

Visualización

Visualizar las señales que se están capturando es un aspecto de vital importancia para que los especialistas decidan si estas cumplen los requerimientos de calidad necesarios.

El primer requisito de esta unidad es la posibilidad de *visualizar en tiempo real las señales* que se capturan. Para esto el mecanismo que se utilice debe tratar de refrescar la pantalla lo menos posible. Una posibilidad de implementación es utilizar la técnica del barrido, es decir ir dibujando líneas y solo cuando se llegue al final de la pantalla (en las x) limpiar y comenzar de nuevo en la primera posición.

Además se debe poder *mostrar señales previamente capturadas* y hacer operaciones comunes como zoom en las y, paneo en las x, y seleccionar los canales que se muestran.

Registro

La unidad funcional de registro es la encargada de controlar el las operaciones de captura. Su primer requisito es el de *registrar estudio a partir de un protocolo*¹ previamente diseñado. Esta funcionalidad es la de mayor importancia en toda la aplicación, ya que es su responsabilidad coordinar las tareas de estimulación, captura y visualización. La ejecución de cada prueba (estímulo) dentro del protocolo tendrá un pequeño espacio temporal entre cada una de estas y se notificará al sujeto cuando está a punto de comenzar una nueva.

En el registro el especialista también podrá *pausar*, *reanudar* o *detener* el registro a su discreción. Al *reanudar* un registro, si la prueba pausada actualmente es incompleta (no transcurrió el tiempo completo de duración) se descartará esta y se comenzará de nuevo. En el caso de *detener* se descartará la prueba actual si esta fue incompleta.

¹ El protocolo se definirá en un sub-epígrafe debajo

El resultado del registro de las señales se almacenará en un fichero binario en una ruta previamente definida donde se encuentren todos los que maneja la aplicación. Este formato binario deberá ser definido por el equipo de desarrollo.

En el epígrafe de arquitectura se discute una propuesta para garantizar que todas las unidades funcionales involucradas en este proceso se realicen en armonía y de la forma más eficiente posible.

Gestión de Estudios

A los estudios que salen como resultado del proceso de registro deberán poder almacenarse para su posterior procesamiento por parte de aplicaciones como el *NSEog*. Debido a esto se hace necesario que la aplicación implemente los siguientes requisitos:

Almacenar estudio: La aplicación deberá ser capaz de almacenar las señales generadas (estímulo) y capturadas (canales registrados) junto a datos del estudio como el nombre, la fecha de registro y notas. Además se almacenarán los datos del protocolo de estimulación utilizado y los datos del sujeto al que se le efectúa el registro.

Listar estudios registrados: La aplicación deberá contar con una base de datos en el sistema de archivos local donde se encuentren los estudios previamente registrados. La aplicación deberá tener alguna interfaz donde muestre el listado de estudios disponibles y además la posibilidad de filtrarlos por nombre y apellidos del sujeto, por clasificación o por contenido de las notas.

Abrir estudio registrado: La aplicación deberá ser capaz de abrir un estudio previamente registrado.

Eliminar estudio: Eliminar un estudio disponible en la base datos local.

Exportar estudio: Exportar un estudio disponible en la base de datos local a una fuente externa.

Importar estudio: Importar un estudio exportado por la aplicación a la base de datos local.

Establecer los datos del sujeto: Cada estudio deberá contener datos del sujeto al que se le efectúa el registro: nombre, apellidos, fecha de nacimiento, clasificación, notas. Esta operación podrá efectuarse antes o después del registro del estudio.

Los metadatos de los estudios registrados deberán ser almacenados en formato JSON en un fichero local denominado algo así como **studies.json**.

Gestión de Protocolos

Los protocolos es la unidad que define la plantilla de estimulación de un estudio. Es decir contiene las pruebas (estímulos) que se le mostrarán a un sujeto para registrar su respuesta. Cuenta con datos como su nombre y notas relevantes de este.

Cuenta con tres requerimientos, *crear o editar protocolo, listar los protocolos existentes y eliminar protocolo*. Dentro de la edición del protocolo se contarán con las opciones de añadir o eliminar estímulos previamente registrados.

Los datos de los estudios registrados deberán ser almacenados en formato JSON en un fichero local denominado algo así como **protocols.json**.

Gestión de Clasificaciones

Las clasificaciones son una funcionalidad que nos permitirán agrupar los estudios según algún criterio determinado. Por ejemplo serán útiles para asignar determinados estudios a investigaciones médicas específicas que facilitarán luego su consulta. Cuenta al igual que los protocolos con los datos de nombre y notas.

De forma análoga a los protocolos cuenta con tres requerimientos: *crear o editar clasificación, listar clasificaciones existentes y eliminar clasificación*.

Los datos de las clasificaciones registradas deberán ser almacenados en formato JSON en un fichero local denominado algo así como **classifications.json**.

Configuración

La aplicación deberá contar con una interfaz que permita configurar los principales aspectos necesarios de esta. Contará básicamente con 4 sub-módulos de configuración: Estimulador, Visualización, Registro y Almacenamiento. A continuación se declaran los aspectos configurables de cada uno de estos.

Estimulador: Tamaño de la pantalla (cm) y resolución de la pantalla (pixels). En ambos casos se tendrán en cuenta tanto la dimensión horizontal como la vertical.

Visualización: Canales a visualizar por defecto (estímulo, horizontal, vertical) y la escala temporal por defecto (ms). La escala temporal es la cantidad de milisegundos que se visualizarán durante un barrido de la pantalla.

Registro: Puerto utilizado para la comunicación con el amplificador. Para esto se deberán listar todos los puertos seriales disponibles en el ordenador para la selección del correcto.

Almacenamiento: Ruta local de almacenamiento. Al seleccionarse una ruta de almacenamiento si esta no cuenta con la estructura necesaria se deberá crear una nueva o abrir la nueva base de datos en caso contrario. Además se tendrá la posibilidad de clonar una base datos existente

localmente a una nueva ruta seleccionando qué aspectos copiar de la original, es decir estímulos, protocolos, estudios, y clasificaciones.

En caso de utilización de Qt como framework de desarrollo se deberá utilizar el mecanismo de los QSettings para la persistencia de las configuraciones.

Requisitos Técnicos

Existen una serie de requisitos que aunque no tienen que ver directamente con la interacción con el usuario son de vital importancia para el correcto funcionamiento de la aplicación. En los siguientes epígrafes se detallan los más importantes.

Adquisición

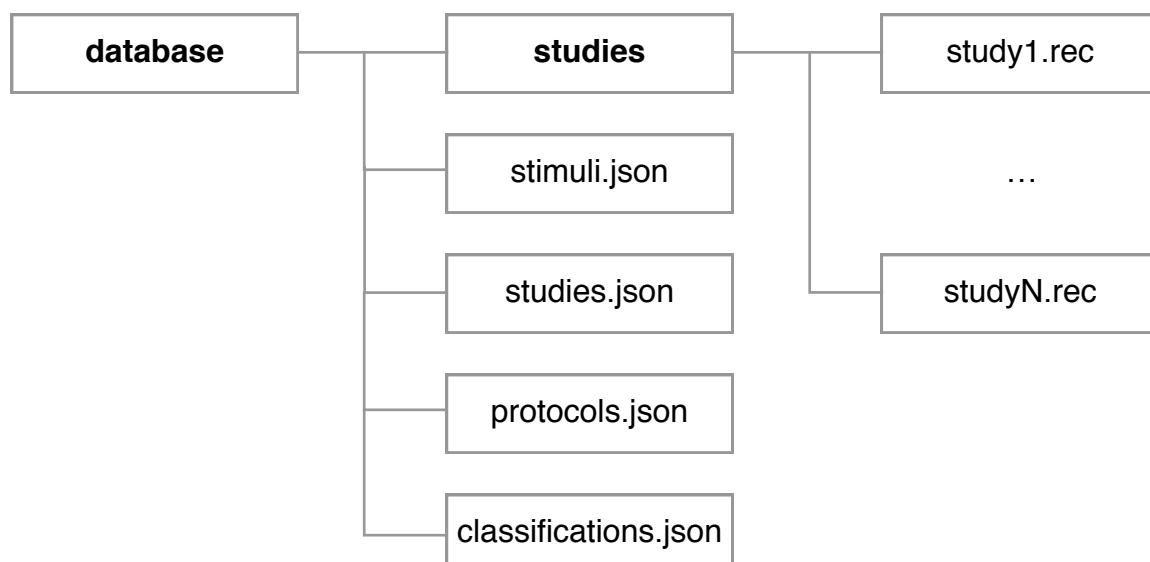
La adquisición se realizará utilizando el *NSNyst Amplifier*, por lo que es importante trabajar en base a las capacidades técnicas del mismo. Para esto se deberá desarrollar un firmware para la plataforma Arduino Due que sea capaz de registrar la información proveniente de los electrodos conectados al sujeto y su envío al ordenador. Este firmware deberá contar con los siguientes requisitos:

- Frecuencia de muestreo a 1 KHz.
- Tamaño de palabra 12 bits.
- Utilizar las interrupciones del reloj del Arduino para la captura. Esto garantizará el muestreo uniforme.
- Envío de las datos al ordenador mediante una interfaz serial, ya sea USB o Bluetooth.
- Envío de los en paquetes de un tamaño que no excederá al del buffer serial del Arduino (64 bytes). Esto garantizará que no se pierdan muestras en la captura.

Se deberá dejar claro el protocolo de comunicación con el ordenador, es decir un juego de comandos para las principales acciones del equipamiento como inicio del registro, finalización del registro, envío de paquetes, etc. Además es necesario implementar un API para el registro de las señales en el lado del ordenador.

Almacenamiento

Se contará con una base de datos local que tendrá tentativamente la siguiente estructura:

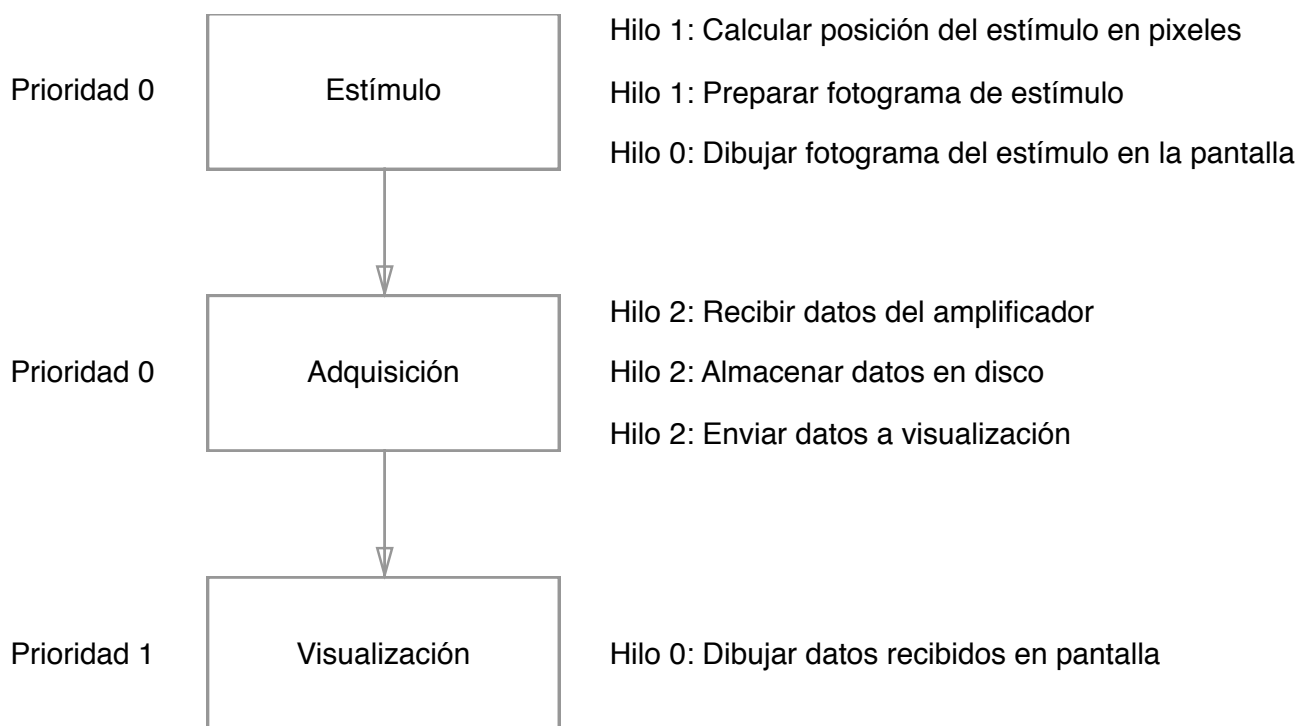


Como se declaró anteriormente en el epígrafe de configuración deberán existir rutinas capaces de crear esta estructura, hacer consultas dentro de esta y poder clonar sus datos a una nueva ruta. La aplicación solo podrá trabajar con una sola de estas bases de datos a la vez, por lo que todas las operaciones que se hagan deberán realizarse sobre la bases de datos activa declarada en la configuración. Se recomienda separar el almacenamiento como unidad funcional independiente para facilitar su desarrollo.

Además se deberá definir el formato binario con que se almacenarán internamente cada uno de los estudios registrados.

Proceso de Estimulación-Captura-Visualización

El proceso que involucra la estimulación, la captura de los datos enviado por el amplificador y la visualización de los mismos en tiempo real es el más complejo de todos los realizados por la aplicación. Debido a esto a continuación proponemos una primera aproximación de la secuencia que debe gobernar este proceso.



La figura anterior representa una propuesta de secuencia de captura para la aplicación. Como se observa se utilizan 3 hilos para tratar de que procesos como la generación del estímulo y la adquisición se realicen de la forma más concurrente posible. De todas maneras la presentación del estímulo debe ocurrir algún tiempo anterior a la adquisición.

Este esquema es una propuesta y muy probablemente sufra cambios, pero al menos es una primera aproximación.

Tareas Iniciales

1. Discutir el proyecto en el seno del grupo.
2. Selección de tecnologías y herramientas de trabajo.
3. Creación de repositorios y otras tareas de soporte.
4. Modelar el dominio de la aplicación.
5. Proponer arquitectura de la aplicación y su modularización.
6. Asignar módulos específicos a los miembros del equipo de trabajo.
7. Diseñar maquetas de todas las interfaces de usuario de la aplicación.