

Laboratorios Digitales: digitalización de un laboratorio académico de física

Autor: Juan Fernando Castañeda Gutiérrez

Asesor: Harold Castro Barrera

Departamento de Ingeniería de Sistemas y Computación

Facultad de Ingeniería

Universidad de los Andes

Tabla de contenido

Tabla de contenido	2
Introducción	3
Descripción General	3
Antecedentes	3
Identificación del problema y de su importancia	3
Objetivos	4
Diseño y especificaciones	4
Especificaciones	4
Restricciones	4
Desarrollo del Diseño	4
Recolección de Información	4
Alternativas de diseño	5
Tecnologías	6
Implementación	6
Backend web	7
API	7
Abrir programa y Abrir rayos X	7
Ingresar parámetros y resultado parámetros	7
Exportar resultados	9
Lógica	10
Modelo relacional	10
Endpoints	11
Front	11
Autenticación	12
Laboratorios experimentales	12
Información	13
Reservar	13
Computar	14
Validación	16
Configurar el proyecto para iniciarlo	16
Métodos	17
Validación de resultados	17
Conclusiones	19
Discusión	19
Trabajo futuro	19
Referencias	20

Introducción

Este proyecto está enmarcado dentro de la iniciativa de la universidad de Laboratorios Digitales. Específicamente, laboratorios digitales es una iniciativa que como objetivo principal busca propulsar la investigación en la Universidad de los Andes. Sin embargo, esta también enmarca el apoyar la docencia, desde un nuevo acercamiento a la realización de sus laboratorios prácticos: el digital. Para esto, se investigó en campo acerca de la digitalización de los laboratorios de la universidad. Y se concluyó desarrollando el diseño, la implementación y las pruebas de la digitalización de un laboratorio, todo dentro de una hipotética plataforma de laboratorios digitales.

Descripción General

Antecedentes

La transformación digital es un tema de prioridad en todas las organizaciones. Su objetivo es adoptar nuevas maneras de trabajar para entregar servicios que se centren en el usuario ante la cambiante tecnología, competencia y necesidades de la población. Específicamente, en las instituciones de educación superior su foco principal es redefinir servicios educativos y procesos operacionales. Según Šereš et al. (2018), los procesos principales en las instituciones de educación superior son los siguientes:

Learning and teaching process	Research process
• study programme accreditation	• research planning
• teaching process preparation and realisation	• research preparation
• teaching process outcomes monitoring	• research conduct
• teaching process assessment	• research outcomes monitoring
• student and teacher mobility realisation	• research evaluation
Enabling processes	Planning and governance processes
• student administration services	• organization management services
• library services	• change and business process management
• staff provision and development services	• plan development
• finance and accounting services	• budget and funds planning
• marketing, sale and distribution services	• performance assessment
• procurement services	

Figura 0. Procesos principales de las IES.

Identificación del problema y de su importancia

La Universidad de los Andes, reconociendo esta necesidad de transformación digital, formuló el proyecto de Laboratorios Digitales específicamente para los procesos de investigación. El objetivo inicial de este proyecto es el de “[d]iseñar una plataforma de laboratorios digitales de alto desempeño, que permita optimizar el uso de recursos tecnológicos ofrecidos y administrados por la Dirección de Servicios de Información y Tecnología en los procesos de investigación (...)” (Uniandes, 2021). Sin embargo, este proyecto puede ser extendido a la transformación digital de todos los laboratorios de la universidad, tanto de investigación como de docencia, a través de su digitalización o

virtualización. Es como consecuencia de lo anterior que se definen los objetivos para este proyecto.

Objetivos

- Caracterizar los laboratorios de investigación y docencia de la universidad.
- Realizar un piloto de digitalización o virtualización para un laboratorio en específico.

Diseño y especificaciones

Especificaciones

El piloto está contemplado como una de dos opciones. La primera es la de digitalizar un laboratorio físico, cuya solución debe consistir en un diseño que contemple una implementación. La segunda es la de digitalizar un laboratorio virtual, cuya solución debe consistir en una arquitectura que solucione las limitaciones del laboratorio y permita masificarlo. Para cualquiera de los dos casos, la solución debe tener en cuenta la experiencia de usuario como una prioridad, y cómo a partir de ella se genera valor a la universidad. Específicamente en el caso del laboratorio físico, la implementación debe intentar acercarse lo mayor posible a la experiencia que tendría un estudiante actualmente en sus prácticas regulares.

Restricciones

El piloto debe poder estar enmarcado en una solución escalable, en el sentido de que su concepto debe ser aplicable a varios laboratorios.

Desarrollo del Diseño

Recolección de Información

El proyecto empezó con una fase de investigación del estado de los laboratorios, tanto investigativos como académicos, en la Universidad de los Andes. Esta se hizo entrevistando a las personas involucradas con los laboratorios: técnicos, asistentes graduados, administrativos y unos pocos estudiantes, y se hizo especial énfasis en la posibilidad de digitalización de los laboratorios. Asimismo, dado que el objetivo era capturar el funcionamiento de los laboratorios como un todo, los laboratorios se eligieron tratando de diversificar los departamentos. Los departamentos en que se hicieron entrevistas fueron los siguientes:

- Microbiología.
- Física.
- Ingeniería mecánica.
- Ingeniería civil.
- Ingeniería de sistemas.

Además, esto se complementó con el trabajo de investigación del semestre pasado de otro estudiante, que se enfocaba en el estado de la investigación en el departamento de ciencias sociales.

Alternativas de diseño

Producto de la investigación, se decidió digitalizar el laboratorio de física intermedia por las siguientes razones. Primero, tenía a su disposición diferentes máquinas que, además de variar en su temática dentro del área de la física, también variaba su principio de funcionamiento, por lo que había mayor posibilidad de encontrar una digitalizable. Segundo, el personal del laboratorio se mostró receptivo al proyecto y mostró su interés de participar de él. Tercero, en las conversaciones con el personal del laboratorio se reveló que el departamento de física está desarrollando una iniciativa de cursos en Coursera. Dado lo anterior, los laboratorios están apoyando con la creación de un kit casero para distribuir y vender junto a los cursos, luego la digitalización de uno de sus laboratorios podría ayudar en su esfuerzo.

A continuación, se eligió la máquina a digitalizar. Para esto se revisaron varias máquinas dentro del laboratorio, pero en mayor medida se confió en el criterio del personal del laboratorio. Específicamente, ellos se decantaron por una máquina que no necesitara muchos actuadores físicos, pues en su experiencia hubiese complicado tanto el proyecto que en un semestre no hubiera sido plausible desarrollarlo. Además, también se buscó que el objetivo pedagógico del laboratorio no se fuese a ver afectado por su digitalización. Lo anterior, pues en departamentos como Ingeniería mecánica o Ingeniería civil, parte de los objetivos pedagógicos es que los estudiantes se expongan a los principios físicos (o tangibles) de los experimentos en la medida de lo posible. De ahí que se eligiera digitalizar la máquina de rayos X de referencia “Phywe XR 4.0 expert unit, 35 kV”.

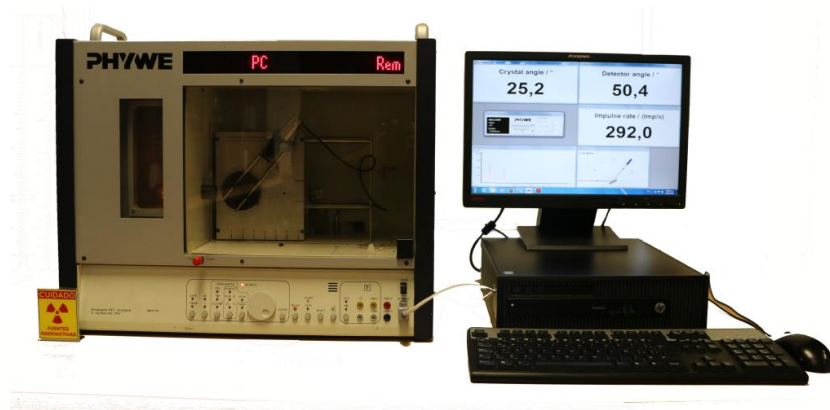


Figura 1. Máquina de rayos x Phywe al lado de su software.

La máquina de rayos X es usada en el laboratorio para estudiar la producción de rayos X para diferentes materiales y condiciones de producción. En cuanto al funcionamiento de la máquina, esta puede ser calibrada físicamente cambiando unos cristales y cambiando el voltaje máximo que se va a trabajar. Sin embargo, esta última configuración siempre se suele dejar en el máximo posible para la máquina por simplicidad: 35kV. Por tanto, aparte de los cristales, todo el resto de configuraciones sobre la máquina pueden ser hechas a

través de un software específico desarrollado por el proveedor de esta. Con lo anterior en mente, se sopesaron diferentes maneras de hacerlo disponible de forma digital.

La primera alternativa implicaba hacer uso de un RDP, el cual consiste en habilitar en el computador la posibilidad de que se conecten a él de manera remota. Sin embargo, esta alternativa fue descartada por la limitación que esta ejercía sobre una de las razones por las cuales se había elegido el laboratorio: la de apoyar el proceso de creación de cursos. Más claramente, los cursos en Coursera son en variadas ocasiones consumidos por personas externas a la Universidad de los Andes. Como consecuencia, utilizar un RPD abriría una vulnerabilidad dentro de la infraestructura de la universidad, pues cualquier externo sería capaz de acceder a un computador que está conectado a la red de esta.

La segunda alternativa fue la de implementar una aplicación web. Esta consistiría en una API que fuera consumida por un backend web, además de un frontend que consumiera a este backend. Esta alternativa, a diferencia de la anterior, implicaba un esfuerzo mayor. Sin embargo, era acorde con todas las razones por las cuales se había elegido el laboratorio. Como consecuencia, fue esta alternativa la elegida.

Tecnologías

En cuanto a la elección de las tecnologías, también se tuvieron en cuenta las razones por las cuales se eligió el laboratorio intermedio. Específicamente, dado el interés del personal del laboratorio por la continuidad del proyecto y su posible expansión a otras máquinas (que, sin embargo, se salen del alcance de este proyecto), se decidió usar un lenguaje de programación con el que estuvieran familiarizados. Por consiguiente, las opciones eran Matlab y Python, que eran los lenguajes que conocían, pero como Python es más adecuado para el desarrollo, se decantó por este en la API y el backend web. En la misma línea de pensamiento, la elección del framework del backend debía estar limitada por la versión de Windows que tenía la máquina con la aplicación. Como actualmente esta cuenta con Windows 7, la máxima versión de Python que soportaba era Python 3.6, así que la elección de FastApi era adecuada para su funcionamiento. Lo anterior teniendo en cuenta que FastApi es un framework web utilizado para construir APIs de manera eficiente en Python.

Por otro lado, en la elección de la tecnología para el frontend y la interacción con la interfaz gráfica de la aplicación de la máquina no se pudo integrar la restricción anterior. En el frontend, debido a la falta de experticia en Javascript, no hubo preferencia por elegir algún framework y se determinó que se iba a usar Svelte con SvelteKit. Mientras tanto, en la interacción de la interfaz se hizo una investigación preeliminar y se decidió que se usaría la herramienta Autoit3. Autoit3 es un lenguaje de scripting diseñado para automatizar las GUI en Windows. Es compatible con varias versiones de Windows, como la que corría el software de la máquina, Windows 7.

Implementación

A continuación, se exponen los diferentes componentes de la implementación, su estado actual de desarrollo y el trabajo futuro que se debe hacer en ellos. El código de ambos componentes se encuentra en los siguientes enlaces para futura consulta.

- <https://github.com/JuanFernandoCastaneda/labs-digitales-back/>
- <https://github.com/JuanFernandoCastaneda/labs-digitales-front/>

Backend web

API

La idea del API es modificar los scripts de Autoit3 que ejecutan acciones sobre la interfaz de la máquina y luego esperar a que este proceso termine para retornar un archivo de Excel. El objetivo de los scripts de Autoit3 es entonces abrir la aplicación, ingresar los parámetros dentro del programa y luego guardar los resultados en un archivo de Excel. Toda la lógica de la API se encuentra dentro de la carpeta “lógica_máquinas/rayos_x”.

Con motivos ilustrativos, la interfaz gráfica de la aplicación de la máquina se muestra en las figuras de esta subsección. Además, vale aclarar que para navegar a través de ella se tomó ventaja de que al presionar la tecla alt se entra en modo de navegación con teclas del teclado. Y como nota adicional, se utilizaron algunos comandos “Sleep” para asumir la (falta de) velocidad del computador a la hora de ejecutar algunas tareas. **En un futuro**, todos estos comandos “Sleep” deberían ser reemplazados por soluciones más precisas que tengan que ver con el estado de la interfaz actual que se tiene en pantalla.

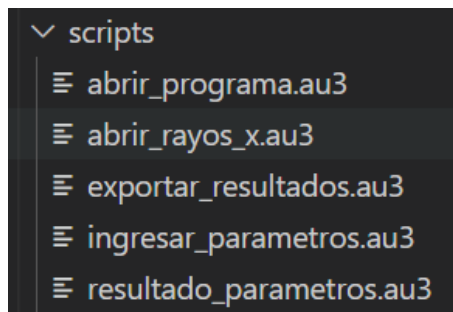


Figura 2. Scripts involucrados en la API.

Abrir programa y Abrir rayos X

En primer lugar, el script de Autoit3 debe esperar a que la aplicación abra con normalidad. Si aparece un mensaje de activación, este debe ser removido antes de seguir con el resto (abrir_programa.au3). Una vez completamente iniciada, se ingresa la combinación “alt” + “e” + “a” para abrir la interfaz de la máquina de rayos X, que se muestra a continuación (abrir_rayos_x.au3).

Ingresar parámetros y resultado parámetros

Muchos parámetros son constantes en los diferentes experimentos, así que únicamente se da la posibilidad de modificar los siguientes: corriente, tiempo, tensión y ángulos. Para ingresar los parámetros a través de la API lo que se tiene es un archivo base llamado “ingresar_parámetros.au3”. Como los archivos .au3 pueden ser abiertos como texto plano, se puede poner en estos unas “banderas” que son reconocidas después por el script en Python. Cuando el script en Python las reconoce, las reemplaza por un texto diferente dependiendo de los parámetros ingresados y las guarda en un nuevo archivo llamado “resultado_parámetros.au3”, que es el finalmente ejecutado.

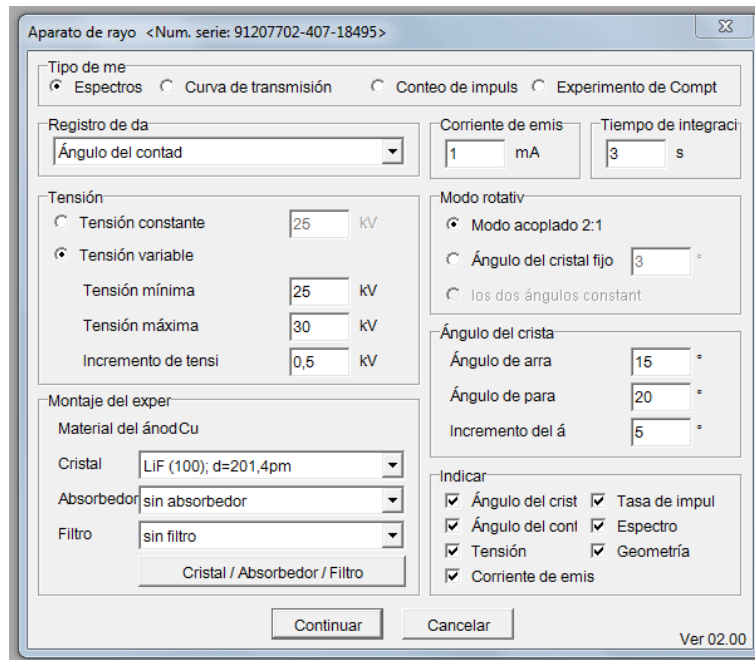


Figura 3. Interfaz de parámetros de rayos x.

En esta interfaz bastaría con navegar a través de ella con “tab” y enviar los valores que ingresó el usuario por parámetro. Sin embargo, la interfaz trata de mantener una consistencia matemática en los valores que se colocan, por lo cual se tuvo que utilizar varios rodeos en aras de evitar comportamientos inesperados. Retomando el proceso, una vez todos los parámetros son ingresados, se llega al botón de continuar y se presiona “enter”.



Figura 4. Interfaz de operación de la máquina de rayos x.

En esta interfaz se muestran los parámetros y cómo van variando a lo largo del tiempo a medida que la máquina va ejecutando la operación. Simplemente hay que presionar “Iniciar medida” y esperar a que finalice la operación. Para estimar el tiempo que va a durar la

operación se usa la siguiente fórmula (en milis), que es una cota superior de lo que se suele demorar la máquina en ejecutar la operación, y que fue extraída de forma experimental:

$$espera = (tiempo + 1.5) * 1000$$

Al parámetro tiempo se le suma 1.5 segundos por los cambios de ángulos.

$$si \text{ anguloIncremento} > 0 \rightarrow$$

$$espera *= ((angulo_parada - angulo_arranque) / angulo_incremento + 1)$$

Se multiplica por la cantidad de cambios de ángulo que hay.

$$si \text{ tensionIncremento} > 0 \rightarrow$$

$$espera += 10000$$

Más mas o menos los 10 segundos que demora recalibrando al ángulo inicial.

$$espera *= ((tension_parada - tension_arranque) / tension_incremento + 1)$$

Multiplicado por la cantidad de cambios de tensión que hay.

$$espera += 20000$$

Se le suman 20 segundos para que las operaciones pequeñas, donde es más el tiempo de recalibrado es mayor que el de la operación, no se vean afectadas.

Si se ingresa tensión constante, la interfaz va a seguir desplegada y se tiene que cerrar manualmente. Una vez se termina la operación, se presiona “Finalizar medida” para pasar a la siguiente interfaz. Si por el contrario se ingresa tensión variable la interfaz se va a cerrar automáticamente. **En un futuro**, se podrían desplegar estos parámetros y su variación en el frontend. Sumado a esto, también se podría optimizar la fórmula de espera para que sea más precisa, o directamente hacer uso de las funcionalidades de Autoit3 para identificar cuando termina la operación. Más precisamente, esperar a que se habilite el botón de “iniciar medida” nuevamente en la interfaz para el caso de tensión constante o esperar a que la interfaz se cierre en el caso de tensión variable.

Exportar resultados

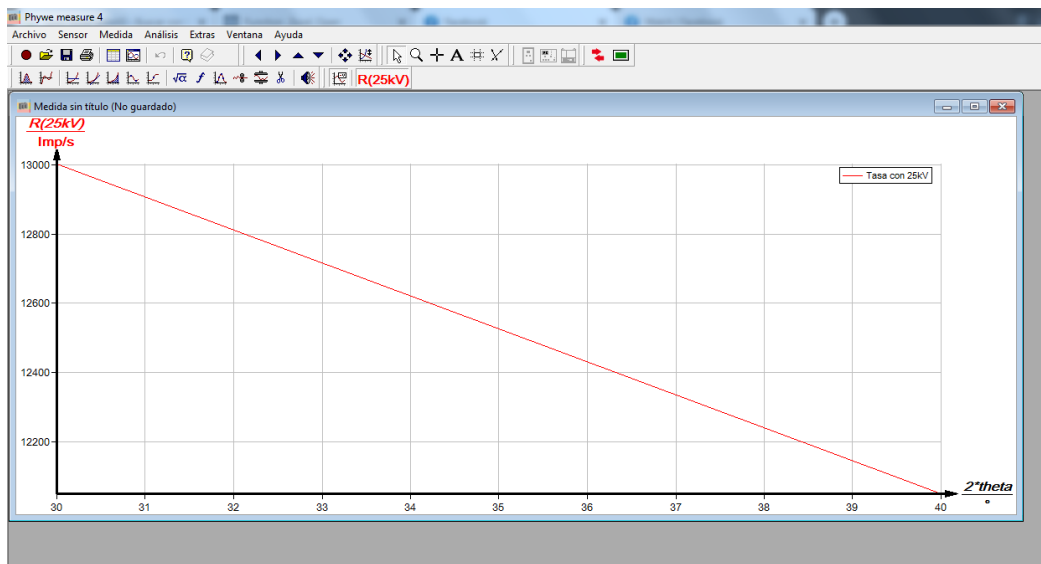


Figura 5. Interfaz del producto de la operación de rayos x

Finalmente, se despliega la gráfica producto. Se puede ingresar la combinación “alt” + “m” + “x” para copiar el contenido a portapapeles. Posteriormente, se usa la librería de Excel de Autoit para abrir un nuevo libro y guardarlo en el mismo directorio del backend (exportar_resultados.au3). Este archivo es el que termina siendo enviado por el backend. Cabe resaltar que la razón por la que se envía un archivo de Excel y no la gráfica es porque en la guía del laboratorio a los estudiantes se les pide que ignoren la gráfica y la hagan ellos mismos en Excel.

Lógica

El backend web se encarga de manejar todas las peticiones del frontend, contiene el modelo de persistencia y el de la autenticación. Por motivos de simplicidad del piloto este fue hecho en el mismo proyecto que la API.

Modelo relacional

A continuación, se describen los significados de las entidades en el modelo.

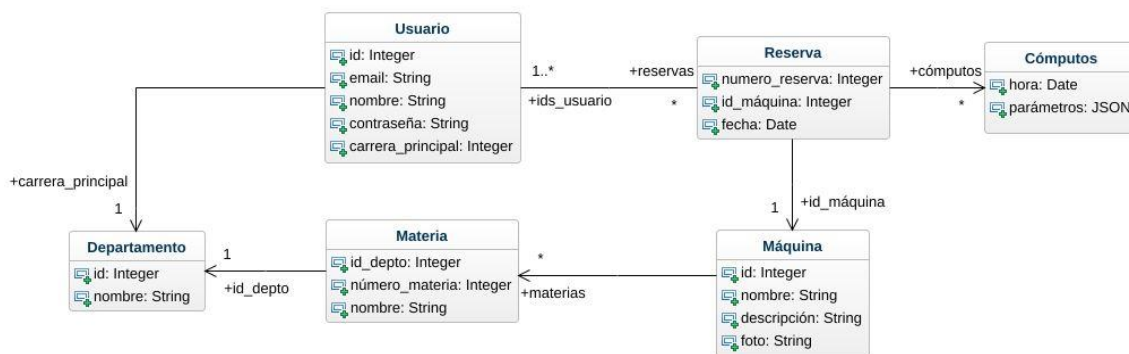


Figura 6. Modelo relacional backend.

Usuario representa a los usuarios que van a usar la aplicación. Esta clase tiene la contraseña almacenada porque la autenticación no está conectada a ningún servicio externo, sino que se tiene un servicio local de JWT. **En un futuro**, sin embargo, esta debería ser eliminada para manejar todo el tema de autenticación con APIs externas.

Departamento representa los departamentos de la Universidad de los Andes. Materia representa las materias que dicta cada departamento (esta no está implementada en el back, por no ser necesaria aún). Actualmente, cuando se retorna la lista de máquinas del back, se hace en orden según su departamento si la persona está logueada. **En un futuro**, se debe considerar cómo implementar al resto de organizaciones para que las “carreras principales” de los usuarios sean consistentes con los departamentos encontrados en la Universidad de los Andes. Además, se debe extraer las materias inscritas para estudiantes de la Universidad de los Andes de Bloque Neón, de tal forma que automáticamente se pueda ordenar también por materias inscritas actualmente.

Finalmente, Máquina representa la información de las diferentes máquinas, no la lógica para ejecutarlas (pues esta se encuentra en la API). En reserva se encuentra la información de todas las reservas y los cómputos que se han generado en ella (representado por la clase

Cómputo). Más claramente, estas cuentan con la información de cuál es la máquina que está reservada y cuáles son los estudiantes que hacen parte de ella (porque pueden reservar en grupo). Para representar esto en el back, es necesario hacer uso de una tabla intermedia many to many entre estudiantes y reserva. Sin embargo, esta lógica no está implementada aún. **En un futuro**, la lógica de las reservas no debería ser implementada por el backend sino por el servicio de Bookeau.

Endpoints

Para el manejo de los endpoints, FastApi tiene a disposición del desarrollador una pestaña “docs”, a la cual se accede con la ruta del backend más “/docs”. En esta pestaña se encuentran todos los endpoints que tiene disponible el back. Más específicamente, se tienen los presentados a continuación.

default	
POST	/token Obtener Token
POST	/maquinas Obtener Maquinas Por Pagina
POST	/maquinas/anonimo Obtener Maquinas Por Pagina Anonimo
GET	/maquinas/{id} Obtener Maquina
GET	/users/me/ Read Users Me

Figura 7. Lista de endpoints que no son de tipo websocket.

- “/token” trabaja lo relacionado con la autenticación, que se hace con un token de JWT.
- “/máquinas” obtiene la lista de máquinas ordenadas cuando el usuario está logueado.
- “/máquinas/anónimo” obtiene la lista de máquinas desordenadas cuando el usuario no está logueado.
- “máquinas/{id}” retorna la información de la máquina que se está consultando.
- “users/me” retorna la información del usuario que está logueado.

Además, hay un endpoint que no aparece representado en “/docs”: “/rayos_x”. Lo anterior, como consecuencia de que es un websocket y no una petición normal. Este endpoint se encarga de ejecutar los scripts del API en orden y de retornar el archivo de Excel. Fue creado como un websocket porque en el frontend se desconocía el tiempo exacto que se iban a demorar las operaciones, además de que podían ocurrir timeouts por su larga duración, así que un websocket fue la solución ideal para mantener la conexión siempre abierta. **En un futuro**, este websocket debe ser mejorado para soportar desconexiones del cliente, de tal forma que se logre recuperar el estado del sistema.

Front

Finalmente, el frontend consiste en las siguientes vistas.

Autenticación



The login form is titled "Iniciar sesión" and is contained within a light gray border. It features two input fields: "Usuario:" and "Contraseña:". Below the password field is a button labeled "Iniciar sesión".

Figura 8. Interfaz de login en el front.

La primera vista es la del login, a la cual se accede con el botón “login” de la barra de navegación. Autentica con JWT. **En un futuro**, esta debería ser capaz de autenticar con la misma base de datos que utilizan otros servicios de la Universidad de los Andes. Más aún, como parte de la visión del proyecto es que apoye los cursos virtuales de Coursera de la universidad, este también debería ser capaz de acceder a los sistemas de autenticación de otras organizaciones para autenticar a los externos. Para lo anterior lo más probable es que sea necesario crear una API pública que puedan consumir las organizaciones que contraten con la universidad de los Andes.

Laboratorios experimentales



The interface for experimental laboratories features a yellow navigation bar at the top with the text "Universidad de los Andes Colombia", "Laboratorios digitales", "Laboratorios experimentales", "Bienvenido, johndoe", and "Cerrar sesión". Below the navigation bar, the main heading is "Laboratorios experimentales". A search bar contains the text "Buscar máquina" and a dropdown menu is set to "Todos los departamentos". Two laboratory cards are displayed: "Pendulo" with a pendulum icon and "Rayos x" with an X-ray machine icon. Each card includes buttons for "Mas información" and "Reservar".

Figura 9. Interfaz de laboratorios experimentales en el front.

La siguiente vista es la de laboratorios experimentales, a la cual se accede con el botón “Laboratorios experimentales” de la barra de navegación. En esta se encuentran las máquinas de experimentos disponibles de la plataforma. Dependiendo de quién sea la

persona que accedió (su departamento y sus reservas) le enseña primero sus reservas y luego las máquinas que tiene su departamento.

Esta vista puede ser accedida por personas que no están logueadas, y de ser el caso, les muestra de manera aleatoria máquinas de cualquier departamento. **En un futuro**, para entregar una mejor experiencia de usuario, se puede almacenar en las cookies la información de la última persona logueada. De esta forma, cuando alguien sin loguear acceda a la plataforma en el mismo dispositivo, se le puedan mostrar las máquinas ordenadas de la misma manera que se le mostrarían a esa última persona logueada. Pero, en caso de que la persona quisiese hacer una acción más importante, como reservar o acceder una reserva, se le solicitaría reloguearse para confirmar que es la identidad correcta.

Dentro de la carta de cada máquina se encuentra el nombre con el que fue registrado, la foto que lo representa y tres opciones: “Más información”, “Reservar” y “Acceder”. Cada una de estas opciones lleva a una vista relacionada con su temática.

Información



Universidad de los Andes
Laboratorios digitales Laboratorios experimentales Login

Información Rayos x

Descripción
Máquina de rayos x para Física I

Figura 10. Interfaz de información de una máquina en el front.

La primera vista, “Información”, muestra una descripción general del funcionamiento de la máquina. **En un futuro**, esta podría mostrar además la información de los parámetros de la máquina, de tal manera que un estudiante que quisiera reservar la máquina pudiese desde antes preparar las operaciones que va a hacer. Además, también podría desplegar el tipo “resultado” que va a tener la máquina, de tal manera que el estudiante también prepare sus procesamientos, por ejemplo de datos, con tiempo.

Reservar

Actualmente el modelo en el backend tiene pensado cómo serían las reservas. Sin embargo, aún no está implementado. **En un futuro**, lo ideal sería hacer uso de Bookeau como sistema manejador de reservas. Esto les daría a los usuarios una interfaz de reservas más completa y con más resiliencia. A cambio, únicamente se tendría que asumir el costo

de la creación de un nuevo centro de reservas, pues Bookeau le cobra a la universidad un precio fijo por una cantidad de reservas que se encuentre dentro de un rango. En cuanto al manejo en la lógica del front, se tendría que hacer peticiones a la API de Bookeau para confirmar las reservas. En comparación, para manejar las reservas grupales (solo en caso de que Bookeau no las soporte) bastaría con hacer una petición a la API de Bloque Neón para confirmar quiénes son los integrantes de un grupo, y luego preguntar a la API de Bookeau si alguno de ellos tiene una reserva en esa hora. De esta manera, cualquiera que fuese parte del grupo podría acceder a la interfaz a realizar operaciones.

Computar

The screenshot shows a web interface titled "Rayos x" with a yellow header. The header contains the logo of Universidad de los Andes, navigation links for "Laboratorios digitales" and "Laboratorios experimentales", and user information "Bienvenido, johndoe" and "Cerrar sesión". The main content area is divided into four panels:

- Parámetros fijos:**
 - Tipo de medida: Espectros
 - Modo rotativo: Modo acoplado 2:1
 - Cristal: LiF
 - Absorbedor: Sin absorbedor
 - Filtro: Sin filtro
- Tensión:**
 - Tensión constante: 30 kV
 - Tensión variable
 - Tensión máxima: 35 kV
 - Tensión mínima: 5 kV
 - Incremento de tensión: 10 kV
- Corriente y tiempo:** (Empty panel)
- Ángulo del cristal:** (Empty panel)

Figura 11. Interfaz de parámetros del cómputo de rayos x en el front.

A la vista computar se accede dando clic desde "acceder". Aquí aparece un menú personalizado según la máquina que se vaya a utilizar. En la primera carta se muestran los parámetros que están fijos en el experimento y en el resto de cartas los que se pueden variar. Más abajo se encuentra el botón computar, que manda la solicitud al backend de que empiece a computar y cambia la interfaz en el frontend. **En un futuro**, el frontend debe ser capaz de validar el input de los usuarios, pues no cualquiera es válido:

- Los números decimales pueden tener máximo un número después de la coma.
- La tensión máxima es 35 kV por limitación de la máquina.
- La corriente máxima es 1 mA por limitación de la máquina.
- La máxima cantidad de veces que puede cambiar la tensión es 15. Por ejemplo: si tengo de tensión mínima 0kV y de incremento de tensión 2kV, puedo poner 30kV como tensión máxima pero no 32kV.
- El ángulo máximo es 55 grados (habría que confirmar esto experimentalmente).

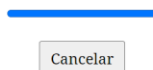
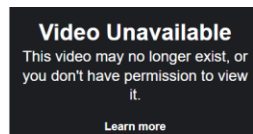


Figura 12. Interfaz de espera del cómputo de rayos x en el front.

En esta interfaz, la cual se accede después de dar clic a “computar”, se puede ver la transmisión en vivo del funcionamiento de la máquina. Actualmente esa transmisión está implementada como un directo de Facebook. Por tanto, hay que utilizar un dispositivo para transmitir en vivo las acciones de la máquina. **En un futuro**, se podría utilizar un dispositivo dedicado para transmitir video, y que sea redirigido a través de una API. Asimismo, también se podría utilizar una grabación en el caso específico de la máquina de Rayos X, pues esta no es muy interactiva en su funcionamiento y es posible que más allá de la primera vez los usuarios no estén interesados. Se sugiere hacer una prueba de validación.

Al mismo tiempo, también se ve la barra de carga de la operación que está haciendo la máquina. Cuando termina la barra de carga, se descarga automáticamente un archivo de Excel con los resultados. Sin embargo, no es posible cancelar la operación y si la conexión se pierde no se puede volver a recuperar. **En un futuro**, se espera que se presente una pequeña interfaz donde se pueda descargar manualmente el archivo, o incluso ver los resultados embebidos en la misma página. Además, se quiere implementar el botón de cancelar, pero este debe estar ligado a poder cancelar los scripts en el backend (una pequeña idea podría ser utilizar otro script de Autoit3 o Bash para cancelarlos). Finalmente, también es ideal que la aplicación sea resistente a desconexiones.

Por otro lado, si un usuario ejecuta un cómputo e ingresa desde la misma cuenta en otro navegador, la aplicación no le presenta el cómputo que ya se está ejecutando, sino que le da la opción de ejecutarlo nuevamente. Un caso similar sucede cuando se reinicia la página, no se retorna al cómputo en ejecución. **En un futuro**, cuando un usuario quiera regresar a la pestaña de computar, se debería presentar el estado del cómputo que ya se está llevando a cabo.

Validación

Configurar el proyecto para iniciarlo

Cuando se va a utilizar el software por primera vez después de haber reiniciado el computador, este siempre fuerza al configurador a abrir y cerrar el cristal de la máquina cuando se va a hacer algún cómputo para asegurar que el cristal esté bien cerrado.

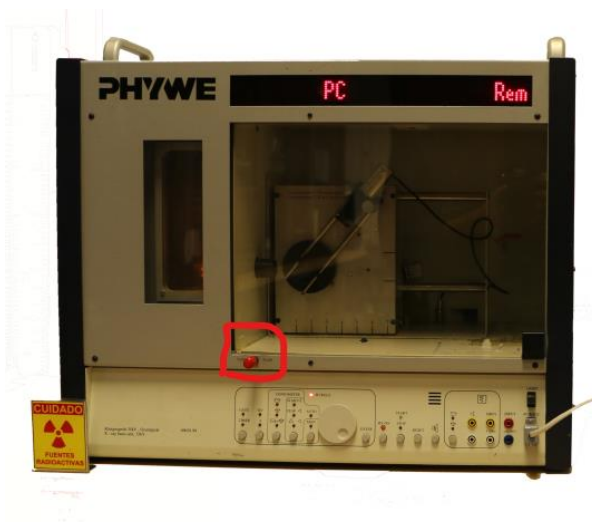


Figura 13. Foto de la máquina donde se señala el botón con el que se abre y se cierra el cristal.

En el proceso de verificar que el software sí reconoció el aseguramiento, se debe intentar correr un cómputo con el software nativo. Si este no genera una ventana de error, sino que ejecuta normalmente, el cristal habrá sido reconocido como asegurado.

En adición a lo anterior, si se reinicia el back se debe asegurar que la última ejecución en el software haya sido con tensión constante, pues el backend y la API dependen de ello para moverse dentro de la interfaz de la máquina. Por simplicidad se pueden hacer esta operación y la anterior en el mismo cómputo: se pone tensión variable y se corre para verificar que el cristal haya sido asegurado.

Después de esto, hay que desactivar todas las notificaciones de la barra de tareas para que estas no interrumpan con la ejecución de los scripts de Autoit. Además, se tiene que asegurar que la pantalla del computador no se vaya a suspender para garantizar que el personal del laboratorio pueda monitorear la ejecución del programa.

Por último, basta con correr sobre una terminal, ubicada en la carpeta del proyecto, el comando “python app.py” para ejecutar el back. Adicionalmente, para exponer los servicios durante desarrollo se utilizó ngrok, pero este último paso solo es una recomendación.

Métodos

Para validar el correcto funcionamiento de la aplicación se crearon diferentes escenarios que cubrían las diferentes combinaciones de parámetros. Sobre todo, se hizo énfasis en los puntos más críticos, que son la tensión variable vs la tensión constante y las variaciones de ángulos. Como tiempo y corriente no representan un reto para los scripts, no se probó la variación de estos (todos los experimentos fueron a 1mA de corriente y 3s de tiempo). A continuación se exponen los escenarios.

1. 35 kV de tensión constante. Ángulo inicial de 3 grados y final de 55, con incremento de a 0.1 grados.
2. 35 kV de tensión constante. Ángulo inicial de 3 grados y final de 55, con incremento de a 1 grado.
3. Tensión variable de 30kV a 35kV en incrementos de 1kV. Ángulo inicial de 5 grados y final de 55, con incremento de a 10 grados.
4. 35 kV de tensión constante. Ángulo inicial de 25 grados y final de 35, con incremento de a 1 grado.
5. Tensión variable de 30kV a 35kV en incrementos de 1kV. Ángulo inicial de 30 grados y final de 35, con incremento de a 1 grado.
6. Tensión variable de 15kV a 19kV en incrementos de 1kV. Ángulo inicial de 10 grados y final de 12, con incremento de a 1 grado.
7. Tensión variable de 15kV a 15.6kV en incrementos de 0.3kV. Ángulo inicial de 10 grados y final de 12, con incremento de a 1 grado.
8. Se pregunto a alguien del personal del laboratorio que lo ejecutara con parámetros consistentes pero al azar.

Validación de resultados

1. El experimento concluyó de manera exitosa.

	A	B
1	Ángulo del cc	Tasa con 35kV
2	$2*\theta/^\circ$	R(35kV)/Imp/
3		
4	6	13110
5	6.2	13085
6	6.4	13103
7	6.6	13078
8	6.8	13071

Figura 14. Resultado del experimento 1, parte inicial.

518	108.8	1457
519	109	1475
520	109.2	1480
521	109.4	1473
522	109.6	1469
523	109.8	1525
524	110	1478

Figura 15. Resultado del experimento 1, parte final.

2. El experimento concluyó de manera exitosa.

	A	B
1	Ángulo del cc	Tasa con 35k
2	$2*\theta/^\circ$	$R(35kV)/Imp,$
3		
4	6	13096
5	8	13082
6	10	12942
7	12	12735
8	14	12350
9	16	11836
10	18	11927

Figura 16. Resultado del experimento 2.

- El experimento no concluyó no de manera exitosa. Después de investigarlo se descubrió que fue porque se pusieron los parámetros mal en el frontend debido a la falta de restricciones (mencionadas en el trabajo futuro del frontend). Específicamente, se intercambiaron los valores de tensión máxima y tensión mínima.

- El experimento concluyó de manera exitosa.

	A	B
1	Ángulo del cc	Tasa con 35k
2	$2*\theta/^\circ$	$R(35kV)/Imp/$
3		
4	50	11793
5	52	12468
6	54	5247
7	56	3376
8	58	2555
9	60	1934

Figura 17. Resultado del experimento 4.

- El experimento concluyó de manera exitosa.

	A	B	C	D	E	F	G
1	Ángulo del cc	Tasa con 30k	Tasa con 31k	Tasa con 32k	Tasa con 33k	Tasa con 34k	Tasa con 35k
2	$2*\theta/^\circ$	$R(30kV)/Imp,$	$R(31kV)/Imp,$	$R(32kV)/Imp,$	$R(33kV)/Imp,$	$R(34kV)/Imp,$	$R(35kV)/Imp,$
3							
4	60	1467	1521	1625	1744	1809	1961
5	62	1222	1268	1363	1481	1513	1641
6	64	1154	1232	1247	1362	1450	1526
7	66	1057	1147	1189	1283	1342	1407
8	68	1048	1091	1123	1250	1312	1334
9	70	1013	1129	1194	1245	1334	1392

Figura 18. Resultado del experimento 5.

- El experimento concluyó de manera exitosa.

	A	B	C	D	E	F
1	Ángulo del cc	Tasa con 15k	Tasa con 16k	Tasa con 17k	Tasa con 18k	Tasa con 19k
2	$2*\theta/^\circ$	$R(15kV)/Imp,$	$R(16kV)/Imp,$	$R(17kV)/Imp,$	$R(18kV)/Imp,$	$R(19kV)/Imp,$
3						
4	20	13810	13821	13781	13752	13644
5	22	13643	13628	13598	13596	13558
6	24	13515	13388	13376	13352	13385

Figura 19. Resultado del experimento 6.

7. El experimento concluyó de manera exitosa.

	A	B	C	D
1	Ángulo del α	Tasa con 15k	Tasa con 15,;	Tasa con 15,6
2	$2*\theta/^\circ$	R(15kV)/Imp	R(15,3kV)/Im	R(15,6kV)/Im
3				
4	20	13809	13824	13848
5	22	13646	13634	13633
6	24	13489	13472	13420

Figura 20. Resultado del experimento 7.

8. El experimento funcionó hasta la parte de generación del Excel, donde fue necesario manualmente reproducir unos pasos del script para que el resto del script pudiera seguirse ejecutando. Hay un bug que hace que en pocas ocasiones la ventana de Excel no aparezca en background y por tanto el script no funciona. **En un futuro**, este bug debe ser resuelto.

Conclusiones

Discusión

Se cumplió con el diseño y la implementación del esqueleto de un proyecto enmarcado en la iniciativa de Laboratorios Digitales. El proyecto está contemplado en medio de la creación de una plataforma web donde se reúnan todos los laboratorios de la universidad, para que puedan ser accedidos por estudiantes y externos de igual manera. Lo anterior generaría nuevas fuentes de ingreso a la universidad y solidificaría la postura de esta como entidad vanguardia en investigación y desarrollo.

Las limitaciones del proyecto vinieron de la mano de la limitación de tiempo. La investigación de campo y el diseño consumieron una cantidad considerable de tiempo que no permitió avanzar más en la implementación. Sin embargo, se espera que ya con estos dos pasos resueltos se pueda concentrar los esfuerzos futuros en la terminación de esta. Muchos de los problemas encontrados derivaron del desconocimiento a profundidad de la herramienta de Autoit3, pues muchas soluciones a los pequeños problemas no eran implementaciones limpias del lenguaje, sino parches sobre el conocimiento que se tenía acerca de este. Para mejorar de manera significativa el proyecto se recomendaría capacitarse bien en la tecnología.

Trabajo futuro

Como resumen del trabajo futuro, parte clave de la continuación del proyecto es indagar acerca de la API de Bloque Neón (para saber si soporta lo aquí pactado, pues no fue posible comunicarse con la DSIT para averiguarlo) e integrar con el servicio de Bookeau. En el frontend falta mejorar mucho de la experiencia de usuario, además restringir los valores que se pueden ingresar por parámetro en la computación y mejorar las restricciones y el uso múltiple del mismo usuario. Por otro lado, en el backend falta implementar la lógica de reservas y en el API corregir el bug que surgió en la validación.

Sin embargo, si se quiere buscar más específicamente, todo lo que es trabajo futuro está incluido en este texto en cada párrafo después de cada combinación de palabras “En un futuro”. Para encontrarlos basta con realizar una búsqueda en el documento con la combinación de palabras anteriores. De esta manera, cada acción del trabajo futuro se encontrará contextualizada en lo que es el estado actual del desarrollo, y de paso será fácil clasificarlo según su categoría (frontend, backend o api).

Referencias

- Departamento de Física de la Universidad de los Andes (2021). *Rayos X*. Laboratorio intermedio.
- Šereš, Laslo & Pavličević, Veselin & Tumbas, Pere. (2018). *Digital Transformation of Higher Education: Competing on Analytics*. 9491-9497. 10.21125/inted.2018.2348.
- Universidad de los Andes. (2021). *Programa de Desarrollo Integral 2021-2025*. Recuperado el 21 de febrero de 2022 de <https://uniandes.edu.co/sites/default/files/asset/document/pdi-2021-2025-final-julio21-baja.pdf>.