Nombre del semillero de investigación

SEMILLERO DE INVESTIGACION EN INGENIERIA DE SOFTWARE

Nombre del proyecto

EXPLORACION Y MEJORA DE VARIAMOS (UNA HERRAMIENTA PARA LA ESPECIFICACION DE MODELOS DE VARIABILIDAD)

Año

2019

Marco del Proceso de ASC: Fortalecimiento o solución de asuntos de interés social.

Este semillero de investigación forma parte de la estrategia de Investigación Formativa de la Universidad EAFIT que ratifica su compromiso con el desarrollo de la Ciencia, la Tecnología, la Innovación y la Creación y lo establece como uno de sus ejes misionales. Con el desarrollo de iniciativas como estas, aportamos a los procesos científico-tecnológicos y creativos que se desarrollan en el seno de su comunidad universitaria y cómo estos permean las experiencias de aprendizaje de los estudiantes.

En el propósito de cultivar talentos y vocaciones científicas y creativas el rol del profesor es vital, ya que es quien desde su pasión, experiencia y conocimiento acerca a los estudiantes a las técnicas propias del saber científico y diseña experiencias de aprendizaje que aporten a la construcción de procesos investigativos.

En consonancia con los Lineamientos de Investigación Formativa de la Universidad, se describe a continuación los principios que rigieron la ejecución del componente con sus resultados, en términos del fortalecimiento de asuntos de interés social, del presente proyecto:

Objetivos del semillero en clave de fortalecimiento de la práctica educativa

Los propósitos educativos-pedagógicos buscados con el desarrollo del proyecto con el semillero incluyen la aplicación de buenas prácticas en la Ingeniería de Software, la introducción al desarrollo orientado por modelos, la investigación y aplicación de patrones de diseño y arquitectura para mejorar la calidad del software, el uso de metodologías innovadoras como la gamificación para mejorar la enseñanza, y la

promoción de la industrialización de la producción de software. Además, se busca proporcionar una experiencia práctica a los participantes a través de proyectos reales, fomentando así una cultura de investigación y consulta que permita explorar y resolver preguntas relevantes en el campo de la Ingeniería de Software.

Metodología

Estudiar el estado del arte relacionado con la gestión de lenguajes de modelado de sistemas en contextos generales de desarrollo de Software. Identificar problemas en la gestión de lenguajes de modelado de software y la generación de código. Explorar la herramienta VariaMos. Identificar uno de los módulos de VariaMos susceptible de ser mejorado. Desarrollar el módulo. Escoger casos de estudio. Ejecutar los casos de estudio. Evaluar y analizar resultados.

Resultados obtenidos

Documentación de las mejoras aportadas al VariaMos. Documentación de las pruebas realizadas a las mejoras del software. Documentación de propuestas para trabajos futuros.

Descripción del fortalecimiento, la solución o el mejoramiento de la práctica educativa

El proceso de fortalecimiento de prácticas educativas, habilidades y competencias en los participantes en el proceso de investigación formativa (semillero) se realizó mediante una combinación de capacitación teórica y práctica. Inicialmente, se presentó información sobre fundamentos de la Ingeniería de Software y metodologías ágiles, proporcionando una base teórica sólida. Posteriormente, los semilleristas participaron en un proyecto real donde aplicaron estos conceptos, fomentando habilidades prácticas en desarrollo de software, modelado y transformación de modelos. Además, se promovió la colaboración y el trabajo en equipo, así como la investigación y el análisis crítico, lo que fortaleció sus competencias en resolución de problemas y pensamiento analítico.

A partir de estas consideraciones, a continuación, se encuentra la sistematización del proceso.

INFORME FINAL DE PROYECTO

Convocatoria: - Financiación de pequeños proyectos a los Semilleros de Investigación 2019

Proyecto: Exploración y mejora de VariaMos (una herramienta para la especificación de modelos de variabilidad)

Código y Titulo Corto: 913-000023 Exploración y mejora de VariaMos

Actividades:

- Estudiar y comprender el funcionamiento actual de VariaMos.
- Estudiar el estado del arte relacionado con la gestión de lenguajes de modelado de sistemas en contextos generales de desarrollo de Software.
- Identificar problemas en la gestión de lenguajes de modelado de software y la generación de código.
- Desarrollar y/o mejorar un módulo de VariaMos.
- Definir caso de estudio.
- Proponer estrategias para probar el desarrollo con el caso de estudio, y ejecutarlas.
- Documentar caso de estudio.

Comentarios:
En el proceso se aplicaron buenas prácticas en la Ingeniería de Software y se utilizaron patrones de diseño y arquitectura para mejorar la calidad del software. Se proporcionó a los participantes del semillero experiencia práctica a través de un proyecto real, fomentando una cultura de investigación y consulta para explorar y resolver preguntas relevantes en este campo. El semillero combinó capacitación teórica y práctica. Inicialmente, se impartieron fundamentos de la Ingeniería de Software y metodologías ágiles, proporcionando una base teórica sólida. Posteriormente, los participantes del semillero se involucraron en un proyecto real donde aplicaron estos conceptos, desarrollando habilidades prácticas en desarrollo de software, modelado y transformación de modelos. Se promovió la colaboración y el trabajo en equipo, así como la investigación y el análisis crítico, fortaleciendo sus competencias en resolución de problemas y pensamiento analítico. Además, los participantes del semillero colaboraron con un estudiante de maestría, lo que les permitió beneficiarse de su conocimiento avanzado, recibir mentoría directa y obtener perspectivas más profundas en el campo de estudio.

Resultados:
Los documentos anexos: Entrega 1 – Overview, Entrega 2 - Overview, Entrega 3 – Overview, Entrega 4 – Overview y Entrega 1 – Overview dan cuenta de las actividades realizadas por los integrantes del semillero.

Entrega 1

Last updated by | Abraham M. Lora | 7 feb. 2020 at 23:47 GMT-5

1. Introducción

En el presente documento se presenta el planteamiento de una problemática general donde se busca resaltar la importancia de un modelado o diseño de soluciones en arduino o sistemas similares antes de directamente programar.

En la actualidad las herramientas que permiten realizar simulaciones en tiempo real de micro-controladores Arduino como Tinkercad suelen presentar apartados donde los desarrolladores deben realizar un diseño del circuito y también una programación del mismo en lenguaje C, sin embargo no presentan apartados para el modelado de los estados o compartimientos que tendrá el proyecto, por lo cual los problemas inminentes es que por culpa de un desarrollo prematuro los programadores suelen desconocer y restarle importancia al dominio del problema que plantean así como también evitan realizar un mejor diseño de una solución bien optimizada lo cual vale mucho la pena antes de codificar.

Moduino busca solucionar todos estos problemas, como su nombre lo indica **Mod** *(Modelar)* y **Uino** *(Arduino)*, será un proyecto enfocado en proveer herramientas dedicadas al desarrollo de diagramas de circuitos, estados y de comportamiento por medio de flujos, donde el desarrollador no necesitará codificar si no hasta al menos obtener un boceto en código totalmente funcional de lo que quiere realizar.

2. Visión del Producto

2.1 Definición del Problema

En la actualidad durante el desarrollo de proyectos en Arduino es muy común observar que los estudiantes suelen presentar inconvenientes a la hora de codificar.

Partiendo de esta base, según diversos profesores de Ingeniería de Sistemas de la Universidad EAFIT, para ellos, la mayoría de estudiantes que llegan a materias enfocadas a electrónica donde se hace uso de placas de micro-controladores como Arduino donde se tenga que realizar algún tipo de codificación, son reincidentes en tener muchos problemas mientras codifican, tales como, las soluciones a base de prueba y error o la repetitividad de código que se realiza lo cual puede conllevar a errores difícilmente visibles que ralentizan drásticamente el proceso de producción del proyecto. A su vez y según los profesores se debe resaltar que el diseño o modelamiento de un problema es fundamental para su desarrollo, lo que comúnmente no suele ser el factor mas importante sobre todo para los estudiantes que antes de plantear el problema y diseñar la solución tienden a ir a la programación de la solución directamente sin tener totalmente claro como conseguirán esa solución a la que quieren llegar.

2.2 Identificación de Referentes

Teorías y Evolución del Problema

Según Marcell Lip quien es un experimentado desarrollador de Software alemán, en una página de su blog llamada *Plan before coding (why software design is needed?)* menciona que no es profesional y de hecho muy riesgo-so el programar directamente antes de diseñar la solución, según su planteamiento al igual que un ingeniero diseña una casa antes de construirla, un programador debe diseñar y modelar que va a realizar antes de pasar a la codificación. Este pensamiento coincide mucho con el de Gautam Dhameja quien menciona en el apartado de desarrollo del famoso sitio web Medium que es mucho mas importante pensar y re-pensar en una solución, diseñarla y luego codificarla, para él es incluso mucho mas importante diseñar que programar.

Hoy en día los desarrolladores tienen un mal habito de programar directamente lo que para ellos en ese instante es la solución, lo que los conlleva a métodos de prueba y error, o a perder el hilo de un programa, mirar su código luego de unos días y no saber nisiquiera que estaban haciendo. Además no se dan cuenta que significativamente al no pensar en el problema podrían estar haciendo algo no optimo, no seguro o simplemente que no superará las expectativas que se desean alcanzar.

Desde hace muchos años los Ingenieros de Software, Programadores, técnicos e informáticos en general se han preocupado mucho por esto y han intentado dar soluciones para poder modelar los problemas a los que se enfrentan diariamente, para ello se han creado lenguajes de modelado que a día de hoy siguen siendo muy utilizados en muchos contextos, los mas relevantes son:

- UML (Unified Modeling Language)
- FFBD (Functional Flow Block Diagrams)
- DFD (Data Flow Diagrams)
- CFD (Control Flow Diagrams)

- Behaviour Diagrams.
- State/mode Diagrams

Todos estos lenguajes de modelado son de alta relevancia y han sido creados para poder definir de la mejor manera la estructura interna, el funcionamiento, el comportamiento, el dominio del problema y demás factores importantes que permiten con un buen manejo solucionar cualquier problema de desarrollo/software (de manera analítica) sin la necesidad de codificarlo, así mismo, hoy en día se han realizado y se siguen realizando esfuerzos por mejorar aún más esta problemática inicial, tratando de llevarla al punto donde estos diagramas sean aún mas útiles y lleguen incluso al punto de producir un código o boceto que pueda ser útil para el programador, quien vería reflejado su diseño en algo totalmente funcional.

Este último planteamiento se puede evidenciar en la linea de productos llamada VariaMos, quienes han sido un referente a estos esfuerzos de darle una usabilidad y mayor impacto a estos diagramas, volviéndolos funcionales o capaces de generar soluciones totalmente adaptables a lo que el usuario quiere y busca obtener, además, es importante señalar que uno de los enfoques de VariaMos en su versión web de la herramienta ha sido el modelado de estados, comportamientos y control de sistemas basados en Arduino, un gran plus ya que como ya se mencionó estos diagramas pueden ser utilizados para producir un boceto de código funcional de estos diseños.

Competencia Directa e Indirecta:

- Ardublock: Se describe a si mismo como un entorno de desarrollo que por medio de arrastrar y soltar se pueden codificar programas para Arduino.
- Tinkercad Circuits: Aunque tinkercad circuits no pueda generar código a partir de modelados o diagramas, es un referente a nivel mundial para el modelado de circuitos, ya que es reconocida por un diseño único, agradable y muy intuitivo para estudiantes que se interesan por el mundo de los sistemas embebidos.
- Circuito.io: Esta herramienta la cual es muy similar a tinkercad circuits permite simular código de arduino a su vez que dispone al usuario un modelador de circuitos para su solución.

2.3 Usuarios Potenciales

Usuarios:

Moduino se centra en estudiantes de Informática o Sistemas, Electrónica, Diseño de Producto o también aquellos interesados en el desarrollo de Software en general. Este programa puede ser utilizado por cualquier numero de usuarios (depende también del numero de clientes).

Clientes:

Nuestros clientes podrían ser Universidades, Colegios, Escuelas Técnicas o personas particulares cuyo interés es el de utilizar el software para proyectos personales comerciales o no comerciales. A su vez, por cada cliente el programa puede ser utilizado por alrededor de 200 a 2000 usuarios.

2.4 Factor Innovador de la Iniciativa

Innovación:

Nuestra iniciativa busca mejorar el uso que se le dan a los Diagramas de Modelado en un entorno enfocado a Arduino, de tal modo que se vuelvan útiles durante la codificación y que no solo se queden como una plantilla o instrucción que en gran medida es esto lo que hace que los desarrolladores opten por codificar directamente, sobre todo aquellos que carecen de tiempo en la realización de los proyectos. Para ello Moduino será capaz de generar todo el código necesario para realizar las acciones que el usuario indique en base a diagramas de estados y de flujos, además que permitirá al igual que otras herramientas realizar un diseño del circuito el cual utilizará para su proyecto.

Soporte Investigativo:

Uno de nuestros soportes investigativos es el uso que le daremos al generador de código que usa VariaMos para la generación de código C totalmente funcional en Arduino, a diferencia de nosotros las herramientas de VariaMos tienen un enfoque de usabilidad industrial, lo cual impide un fácil entendimiento de estos diagramas para estudiantes (que son los usuarios en los que nos queremos centrar), por eso toda la investigación en cuanto a meta-modelos, análisis de diagramas, producción y relación entre diagramas y código que utilizaremos se lo atribuiremos a dicho proyecto en donde haremos no solo uso si no mejoras en este generador que es relevante para el buen funcionamiento de ambos proyectos.

Tipo de Competencia:

En nuestra área la competencia suele ser muy indirecta, ya que las herramientas similares existen, pero no cumplen nuestro principal objetivo el cual se trata de darle usabilidad a los diagramas modelados.

Sin embargo, y como lo menciona uno de los profesores entrevistados, existen herramientas que permiten generar código (lo cual se podría considerar como competencia directa), pero no son lo suficientemente buenas ya que el enfoque suele ser otro y en algunos casos no son las herramientas si no servicios desarrollados por terceros para esas herramientas, de tal modo que buscan otro tipo de clientes, generalmente enfocados a Industrias.

2.5 Factor Innovador de la Iniciativa

Las características funcionales básicas del proyecto son las siguientes:

- Edición de diagramas de circuitos intuitivos.
- Edición de diagramas de estados.
- Edición de diagramas de flujo.
- Validación de diagramas.
- Persistencia de datos a modo de proyectos.
- Generación de código en Arduino en base a los diagramas.
- Tutoriales enfocados al aprendizaje de la herramienta.

3. Evidencias

Para las evidencias hemos realizado un vídeo donde aparecen 3 entrevistas (por medio de audio), donde realizamos preguntas concretas a 2 profesores de la Universidad y un estudiante de Ingeniería de Diseño de Producto próximo a graduarse.

Para ello todas las entrevistas abordan el problema, donde los entrevistados no saben de nuestra solución o la utilidad que le pensamos dar más allá de lo que ellos conocen, por lo cual aún no se les presenta a ellos el factor innovador de Moduino.

Adicionalmente hemos realizado una entrevista con el Product Owner (Andrés Lopez), en la cual éste explica la utilidad y usabilidad que considera tendrá Moduino y porque es importante en base a su desarrollo con la linea de productos de modelado VariaMos para su trabajo de maestría.

Link del video de evidencias: https://www.youtube.com/watch?v=Y6AZITLTGbc&feature=youtu.be
Link del video (Entrevista al PO): https://www.youtube.com/watch?v=q8TauQMIt6w&feature=youtu.be

4. Canvas

<table>
<tr><td colspan="3">NOMBRE DEL PROYECTO</td><td colspan="4">MOTIVACION</td></tr>
<tr><td colspan="3">MODUINO</td><td colspan="4">CREAR UN SISTEMA DE APRENDIZAJE PARA ESTUDIANTES</td></tr>
<tr><td rowspan="2">RECURSOS HUMANOS PARTICIPANTES</td><td colspan="3">ACTIVIDADES CLAVES</td><td colspan="3">OBJETIVO PRINCIPAL</td></tr>
<tr><td>PREPARACION</td><td>EJECUCION</td><td>PUESTA EN OPERACION</td><td colspan="3">MODELADO DE DIAGRAMAS DE FLUJO, ESTADO Y DOSEÑO DEL CIRCUITO PARA LA GENERACION DE CODIGO</td></tr>
<tr><td>- DESARROLLADOR
- ANALISTA
- DISEÑADOR
- PRODUCT OWNER</td><td>-IDENTIFICAR LAS NECESIDADES Y DIFICULTADES DE LOS ESTUDIANTES DE LOS PRIMEROS SEMESTRES A LA HORA DE DIAGRAMAR Y CODIFICAR
- ENTREVISTAS A ESTUDIANTES Y PROFESORES</td><td>-REALIZAR METAMODELOS PARA LA GENERACION DE CODIGO

-DESARROLLAR E IR ENTREGANDO AVANCES AL PRODUCT OWNER</td><td>- REUNIONES PERIODICAS CON EL PRODUCT OWNER

- CAPACITACION EN LA SINTAXIS DEL CODIGO ARDUINO</td><td colspan="3">OBJETIVOS DE LOS INTERESADOS
- ACCESO AL CODIGO GENERADO
- REALIZACION DE DIAGRAMAS
- UNIFICACION DE TODOS LOS MODELOS Y CIRCUITOS
BENEFICIOS E IMPACTOS
- UTILIZACIÓN DE DIAGRAMAS REALIZADOS
- APRENDIZAJE DE CODIGO ARDUINO
- EFICIENCIA A LA HORA DE REALIZAR PROYECYOS</td></tr>
<tr><td colspan="2">INTERESADOS</td><td colspan="2">ACCIONES Y ACTIVIDADES PARA ASEGURAR LA CALIDAD</td><td colspan="3">QUE LOS APASIONA A PARTICIPAR</td></tr>
<tr><td colspan="2">ESTUDIANTES Y PROFESORES</td><td colspan="2">PRUEBAS DE MODELACION Y CONEXION DE CIRCUITOS</td><td colspan="3">EL RETO QUE ES DESARROLLAR UN PROGRAMA DE APRENDIZAJE Y AYUDA PARA LOS ESTUDIANTES DE LA UNIVERSIDAD EAFIT</td></tr>
<tr><td rowspan="2">FACTORES CLAVES DE EXITO</td><td rowspan="2">RETOS</td><td colspan="4">RIESGOS</td></tr>
<tr><td>DESCRIPICON DEL RIESGO</td><td>PROBABILIDAD</td><td>IMPACTO</td><td>ACCION DE MITIGACION</td></tr>
<tr><td rowspan="3">- GENERACION DE CODIGO AL INSTANTE
- ENTORNO GRAFICO INTUITIVO
- VISUALIZACION DE ERRORES
- ESTRUCTURACIÓN CORRECTA DEL DIAGRAMADO</td><td rowspan="3">- GENERACION DE CODIGO EN BASE A DIAGRAMAS

- DISEÑO INTUITIVO PARA ESTUDIANTES

- VALIDACIÓN DEL CIRCUITO</td><td>FALLAS EN EL GENERADOR DE CODIGO</td><td>MEDIA</td><td>MEDIO</td><td>CORRECCIÓN MANUAL DEL USUARIO</td></tr>
<tr><td>ERRORES DE PERSISTENCIA</td><td>BAJO</td><td>ALTO</td><td>RESPALDO DE PERSISTENCIA</td></tr>
<tr><td>FALLAS EN EL GRAFICADO DEL SISTEMA</td><td>BAJO</td><td>ALTO</td><td>REESTRUCTURACION DEL SISTEMA</td></tr>
</table>

5. Modelo de Dominio

Profesor: Docente encargado de instruir a sus alumnos para desarrollar sistemas de ingeniería mediante arduino que le permitan ampliar sus conocimientos y lograr soluciones de alto nivel.

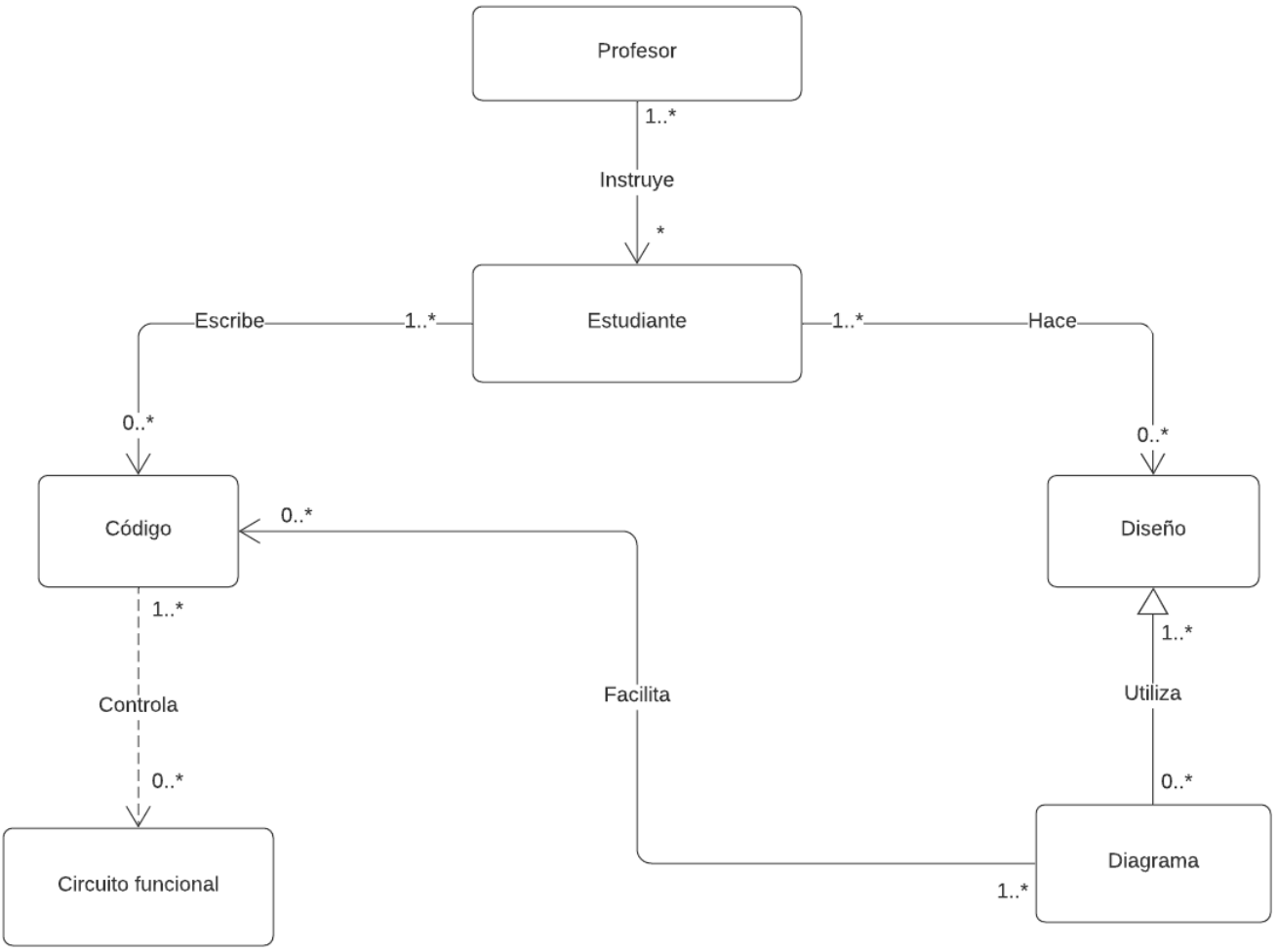
Estudiante: Persona que recibe las instrucciones del docente para desarrollar sistemas en arduino a través de unos pasos de diseño y codificación que le permiten obtener un producto funcional.

Diseño: Elemento gráfico creado por el estudiante para tener una visión mucho más clara de la solución y guiarse de este para que se le facilite el apartado de codificación. No siempre los estudiantes utilizan este elemento de ayuda tan importante.

Diagrama: Es un elemento de diseño que es de gran importancia a la hora de la solución, ya que traduce esta de una manera más simple pero manteniendo sus elementos más importantes a tener en cuenta.

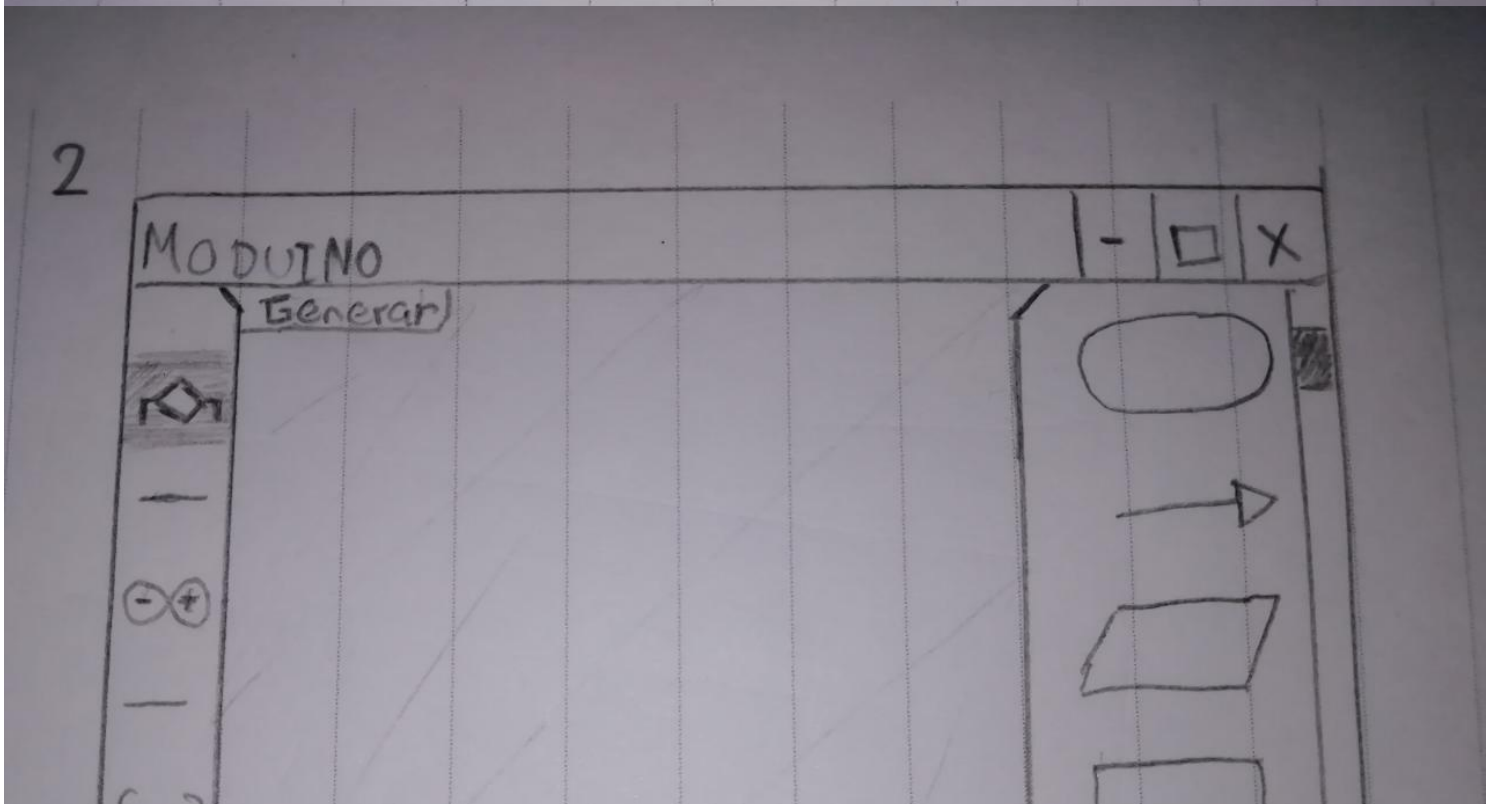
Código: Instrucciones que el estudiante le da a un sistema para que ejecute sus funcionalidades, es el lenguaje que luego es traducido por una computadora para finalmente quedar en el sistema binario que es lo único que entiende cualquier componente de hardware.

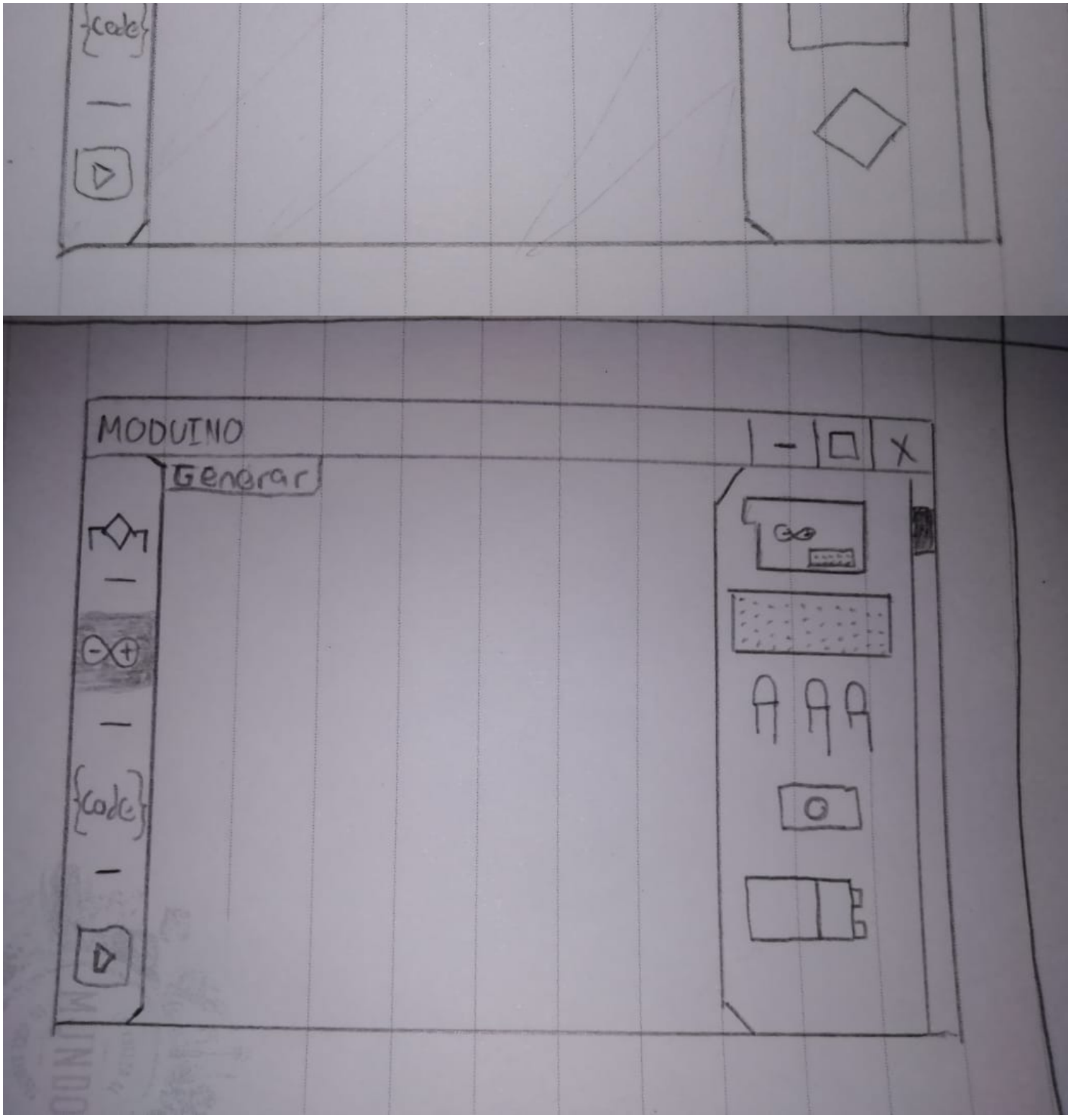
Circuito Funcional: Integración de los componentes físicos, en este caso los compatibles con arduino, que ya dan una solución al problema planteado mediante un funcionamiento controlado por el código.

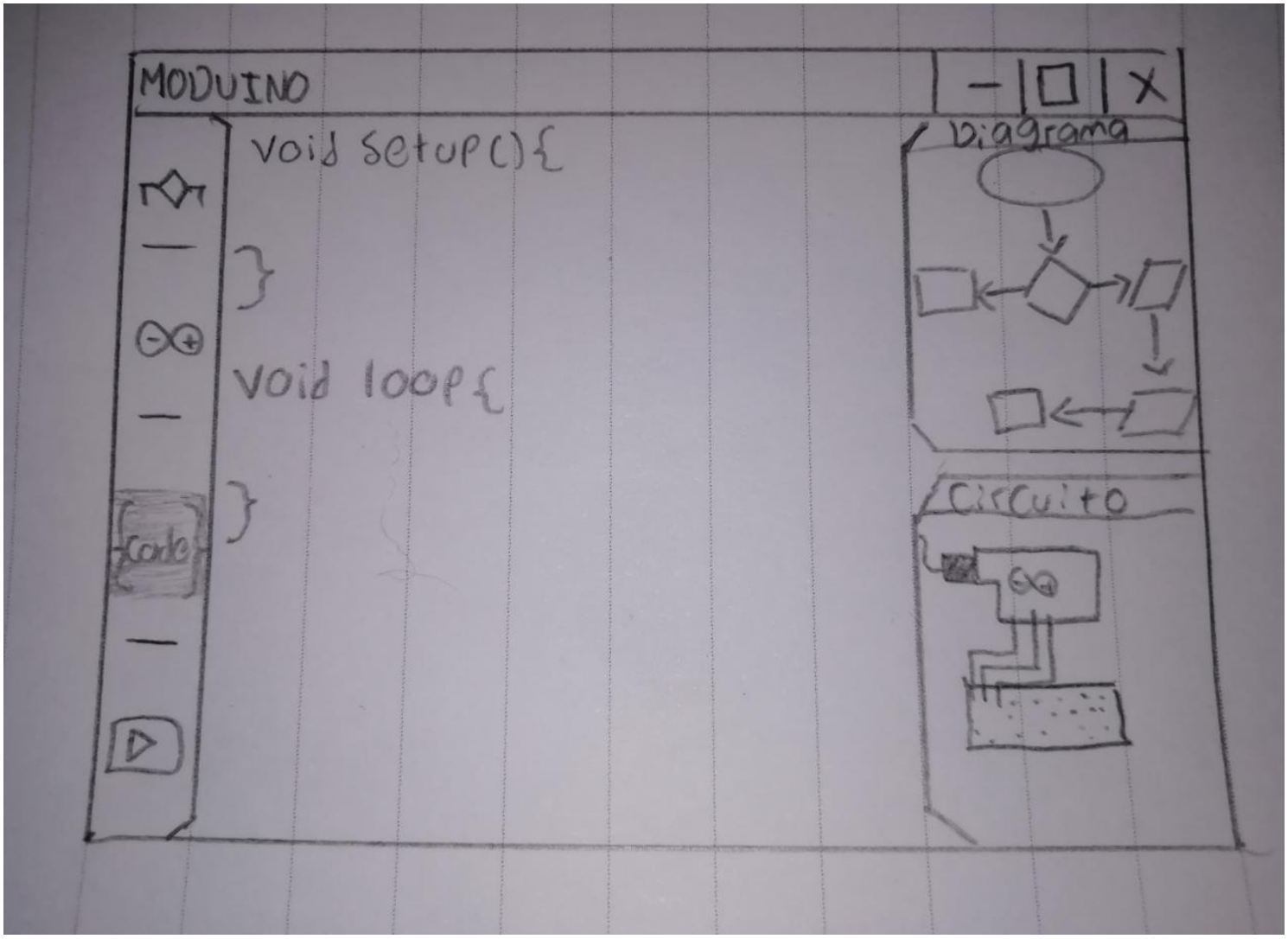

Profesor
1..*
Instruye
*
Escribe
1..*
Estudiante
1..*
Hace
0..*
0..*
Código
0..*
Diseño
1..*
1..*
Controla
Facilita
Utiliza
0..*
0..*
Circuito funcional
Diagrama
1..*

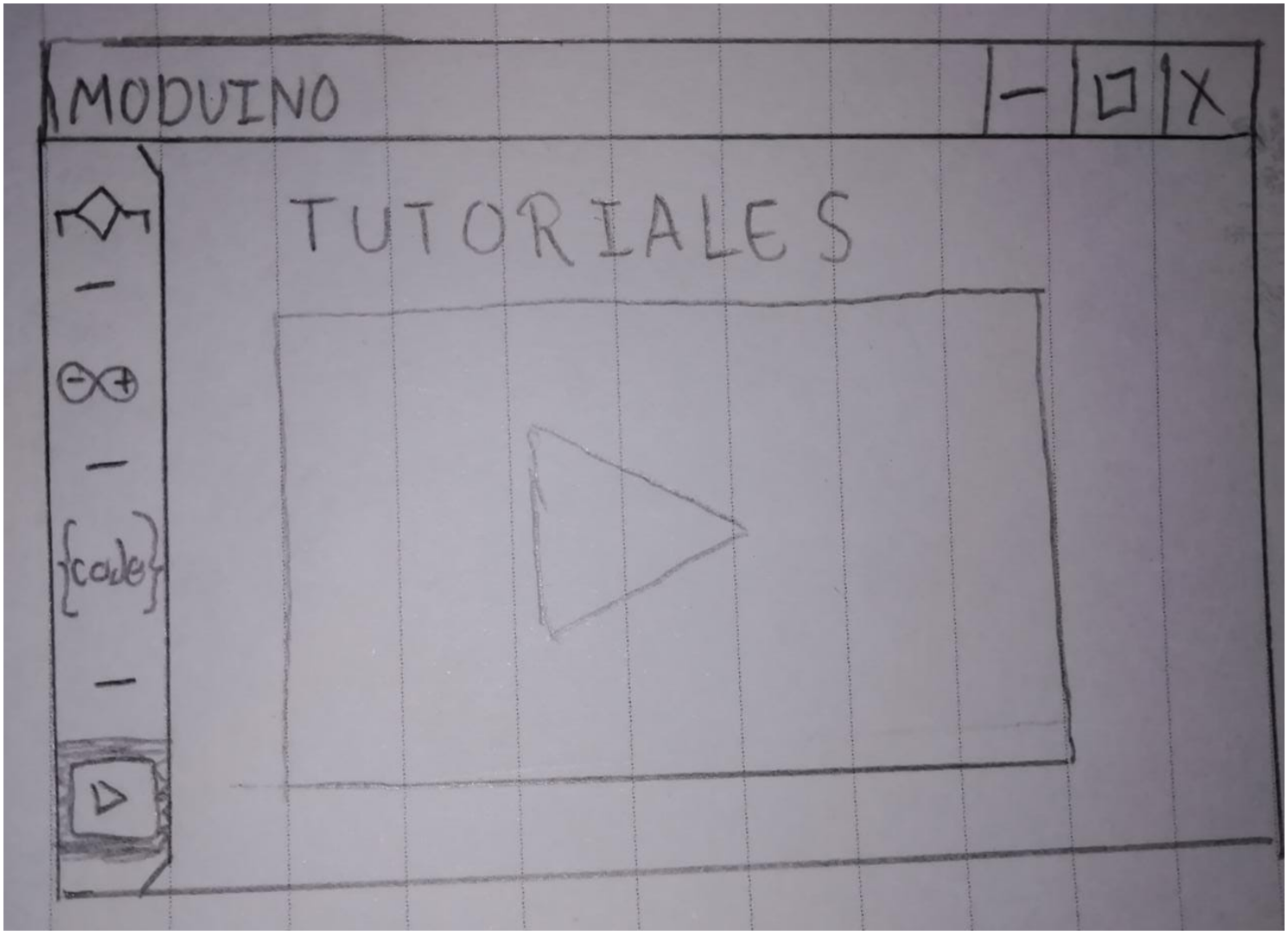
6. Sketches

1
MODUINO
MODUINO
COMENZAR
{code}


2
MODUINO
Generar


{code}
MODUINO
Generar
{code}


MODUINO
- | □ | X
void setup(){
}
void loop{
}
code
Diagrama
Circuito



7. Conclusiones

- Los estudiantes deberían pasar menos tiempo codificando y más tiempo diseñando para lograr mejores soluciones.
- La ingeniería va empezar a evolucionar al punto de no ser tan mecánica a la hora de sólo escribir código, sino que se empezará a centrar en soluciones de más alto nivel a los problemas.
- Arduino es una herramienta fundamental y de aprendizaje para pensar en soluciones a problemas comunes.
- En el mercado hay herramientas que dan una estructura de código para arduino o que codifican comportamiento sin escribir completamente código, pero con herramientas que son difíciles de usar y no cumplen con buenos niveles de usabilidad.
- Si el problema abordado en el presente documento logra resolverse con soluciones óptimas, sería un proyecto de gran impacto y que cambiaría mucho la manera en la que pensamos las soluciones tecnológicas hoy en día.

8. Referencias Bibliograficas

[1] P. (2020). Plan before coding (why software design is needed?). [online] Howtosurviveasaprogrammer.blogspot.com. Recuperado de: http://howtosurviveasaprogrammer.blogspot.com/2018/09/plan-before-coding-why-software-design.html .

[2] Anon, (2020). [online] Recuperado de: https://www.quora.com/Which-are-the-alternatives-to-UML .

[3] VariaMos – Families of Systems & SAS Modeling Tool. (2020). Recuperado de: https://variamos.com/home/.

[4] About VariaMos – VariaMos. (2020). Recuperado de: https://variamos.com/home/about/ .

[5] Ardublock | A Graphical Programming Language for Arduino. (2020). Recuperado de: http://blog.ardublock.com/

[6] Circuit Design App for Makers- circuito.io. (2020). Recuperado de: https://www.circuito.io/

[7] Circuits on Tinkercad. (2020). Recuperado de: https://www.tinkercad.com/circuits

9. Video

En este video presentamos el Pitch, Canvas y algunos Sketches dibujados del proyecto Moduino.

Link del Video: https://youtu.be/LXzOnWC9xp8

Entrega 2

Last updated by | Mateo Montes Loaiza | 14 mar. 2020 at 0:07 GMT-5

1.Introducción

En este documento se presenta el contenido de aspectos muy importantes en el desarrollo del proyecto Moduino. Dando con más detalle y contando con los últimos cambios desarrollados en el proyecto, un enfoque que explique con argumentos porque realizamos unas pruebas de usabilidad a la base ya desarrollada y establecida, mientras que por otra parte tratamos, que cambios se realizaron a la generación de código en base a diagramas en Arduino lo cual es el principal de los objetivos del proyecto. Ligando todo esto a la integración completa en el proyecto de modelado de diagramas Arduino a VariaMos web, el cual es un proyecto ya desarrollado sobre el cual trabajará Moduino para implementar todo lo necesario buscando una mejoría constante en dicho proyecto.

1.1 Enfoques del Proyecto

En Moduino actualmente se trabajan en dos aspectos muy importantes, que ameritan ser tratados de manera unitaria a modo de sub-puntos adicionales para poder enseñar con detalle que es lo que se está haciendo y con que fines.

- **Enfoque en front-end:** En cuanto a front-end Moduino realizó la documentación y planeación de unas pruebas de usabilidad que serán utilizadas para obtener una retroalimentación sobre que tan usable es la plataforma de VariaMos Web en la actualidad, obteniendo de esta manera la opinión de los participantes de las pruebas por medio de encuestas y a su vez estás serán utilizadas para poder idear cambios que puedan mejorar significativamente los tiempos que se tardan las personas en utilizar funciones o acciones de la web.

- **Enfoque en back-end:** En cuanto al back-end Moduino realizó un importante desarrollo con la finalidad de parametrizar componentes Arduino en formato JSON, de tal modo que éste sea leído y procesado para poder generar un código basado en componentes. Esto era algo totalmente requerido e importante puesto que cada circuito de Arduino es único ya que usa sus propios componentes.

2.Diagramas

2.1. Modelo de dominio
Profesor: Docente encargado de instruir a sus alumnos para desarrollar sistemas de ingeniería mediante arduino que le permitan ampliar sus conocimientos y lograr soluciones de alto nivel.

Estudiante: Persona que recibe las instrucciones del docente para desarrollar sistemas en arduino a través de unos pasos de diseño y codificación que le permiten obtener un producto funcional.

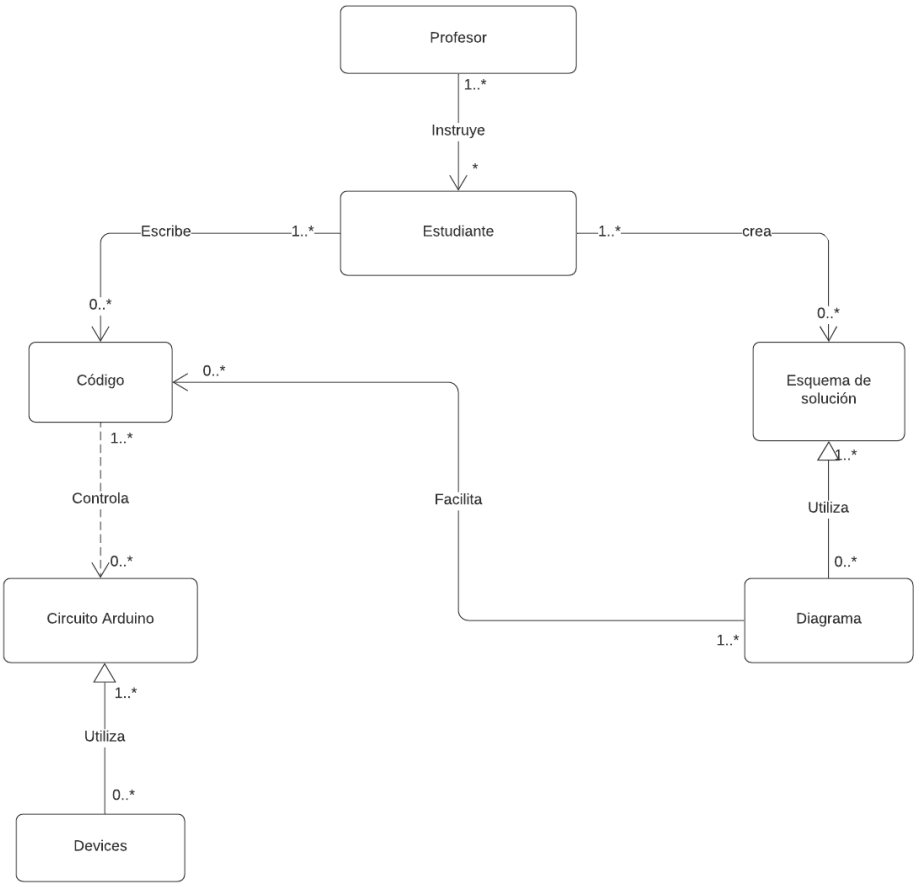
Esquema de solución: Elemento gráfico creado por el estudiante para tener una visión mucho más clara de la solución y guiarse de este para que se le facilite el apartado de codificación. No siempre los estudiantes utilizan este elemento de ayuda tan importante.

Diagrama: Es un elemento de diseño que es de gran importancia a la hora de la solución, ya que traduce esta de una manera más simple pero manteniendo sus elementos más importantes a tener en cuenta.

Código: Instrucciones que el estudiante le da a un sistema para que ejecute sus funcionalidades, es el lenguaje que luego es traducido por una computadora para finalmente quedar en el sistema binario que es lo único que entiende cualquier componente de hardware.

Circuito Arduino: Integración de los componentes físicos, en este caso los compatibles con arduino, que ya dan una solución al problema planteado mediante un funcionamiento controlado por el código.

Devices: Componentes físicos utilizados para integrarlos con arduino y de esta manera darle solución al problema planteado.


Profesor
1..*
Instruye
*
Escribe
1..*
Estudiante
1..*
crea
0..*
0..*
Código
0..*
Esquema de solución
1..*
Controla
Facilita
Utiliza
0..*
0..*
Circuito Arduino
Diagrama
1..*
1..*
Utiliza
0..*
Devices

2.2. Tablero del Proyecto

<table>
<tr><td colspan="2">NOMBRE DEL PROYECTO</td><td colspan="4">MOTIVACIÓN</td></tr>
<tr><td colspan="2">MODUINO</td><td colspan="4">CREAR UN SISTEMA DE APRENDIZAJE PARA ESTUDIANTES</td></tr>
<tr><td>RECURSOS HUMANOS PARTICIPANTES</td><td colspan="3">ACTIVIDADES CLAVE</td><td colspan="2">OBJETIVO PRINCIPAL</td></tr>
<tr><td rowspan="4">-DESARROLLADOR

-ANALISTA

-DISEÑADOR

-PRODUCT OWNER</td><td>PREPARACIÓN</td><td>EJECUCIÓN</td><td>PUESTA EN OPERACIÓN</td><td colspan="2">MODELADO DE DIAGRAMAS DE FLUJO, ESTADO Y DISEÑO DEL CIRCUITO PARA LA GENERACIÓN DEL CÓDIGO</td></tr>
<tr><td rowspan="3">-ENTREVISTAS A ESTUDIANTES Y PROFESORES
-VALIDAR USABILIDAD DE PRODUCTO YA EXISTENTE</td><td rowspan="3">-REALIZAR METAMODELOS PARA LA GENERACIÓN DE CÓDIGO
-DISEÑAR INTERFAZ INTUITIVA PARA EL USUARIO
-DESARROLLAR E IR ENTREGANDO AVANCES EN EL DIAGRAMADO Y GENERACIÓN DE CÓDIGO AL PRODUCT OWNER</td><td rowspan="3">-REUNIONES PERIÓDICAS CON EL PRODUCT OWNER
-CAPACITACIÓN EN LA SINTAXIS DE CÓDIGO ARDUINO
-VALIDAR USABILIDAD E INNOVACIÓN CON EL USUARIO</td><td colspan="2">OBJETIVOS DE LOS INTERESADOS</td></tr>
<tr><td colspan="2">-ACCESO AL CÓDIGO GENERADO
-REALIZACIÓN DE DIAGRAMAS
-UNIFICACIÓN DE TODOS LOS MODELOS Y CIRCUITOS</td></tr>
<tr><td colspan="2">BENEFICIOS E IMPACTOS</td></tr>
<tr><td colspan="2">-UTILIZACIÓN DE DIAGRAMAS REALIZADOS
-APRENDIZAJE DE CÓDIGO ARDUINO
-EFICIENCIA A LA HORA DE REALIZAR PROYECTOS</td></tr>
<tr><td colspan="2">INTERESADOS</td><td colspan="2">ACCIONES Y ACTIVIDADES PARA ASEGURAR LA CALIDAD</td><td colspan="2">QUÉ LOS APASIONA A PARTICIPAR</td></tr>
<tr><td colspan="2">ESTUDIANTES Y PROFESORES</td><td colspan="2">PRUEBAS DE MODELACIÓN Y CONEXIÓN DE CIRCUITOS</td><td colspan="2">EL RETO QUE ES DESARROLLAR UN PROGRAMA DE APRENDIZAJE Y AYUDA PARA LOS ESTUDIANTES D LA UNIVERSIDAD EAFIT</td></tr>
<tr><td colspan="2">FACTORES CLAVES DE ÉXITO</td><td colspan="2">RETOS</td><td colspan="2">RIESGOS</td></tr>
<tr><td colspan="2" rowspan="4">-GENERACIÓN DE CÓDIGO AL INSTANTE

-ENTORNO GRÁFICO INTUITIVO

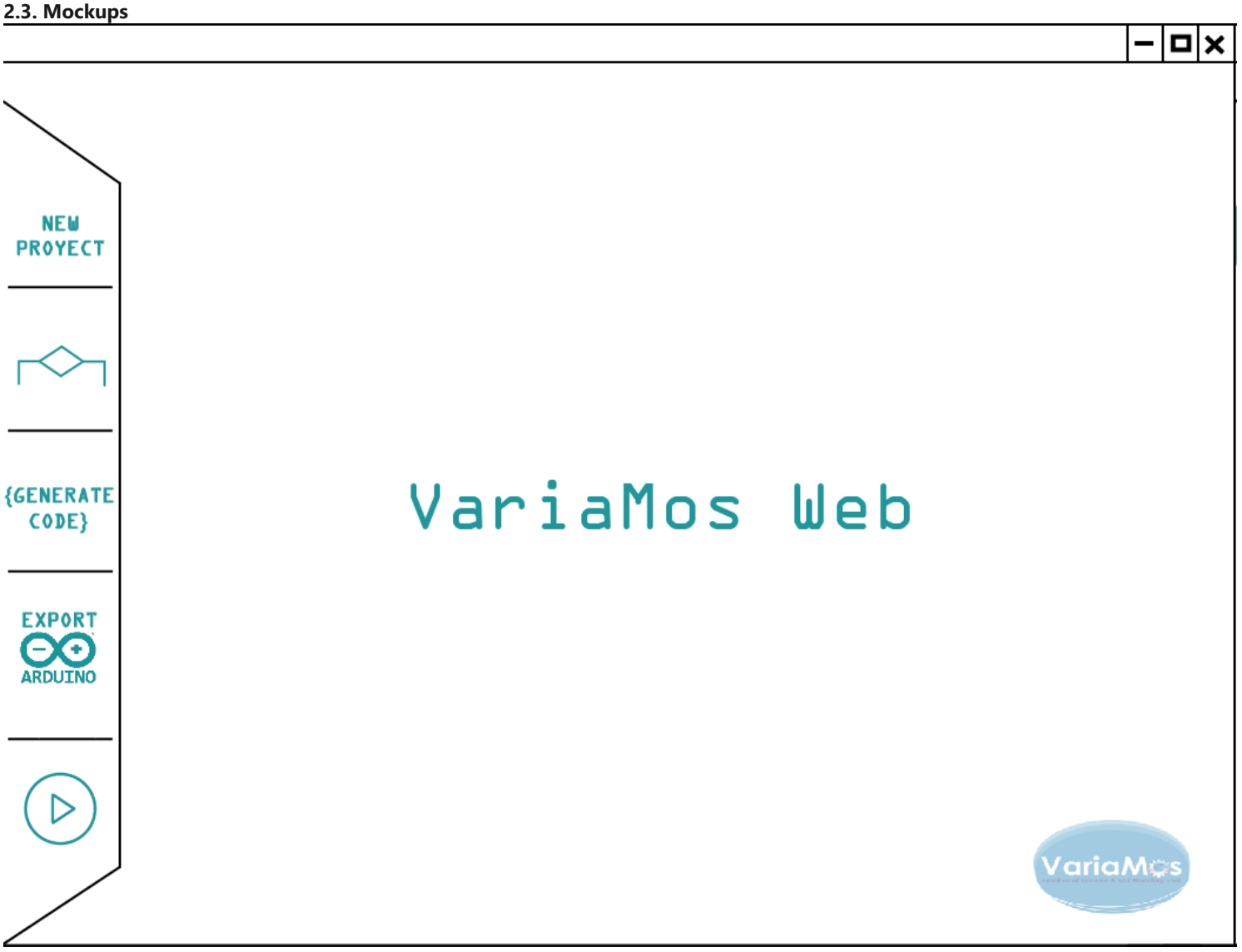
-VISUALIZACIÓN DE ERRORES

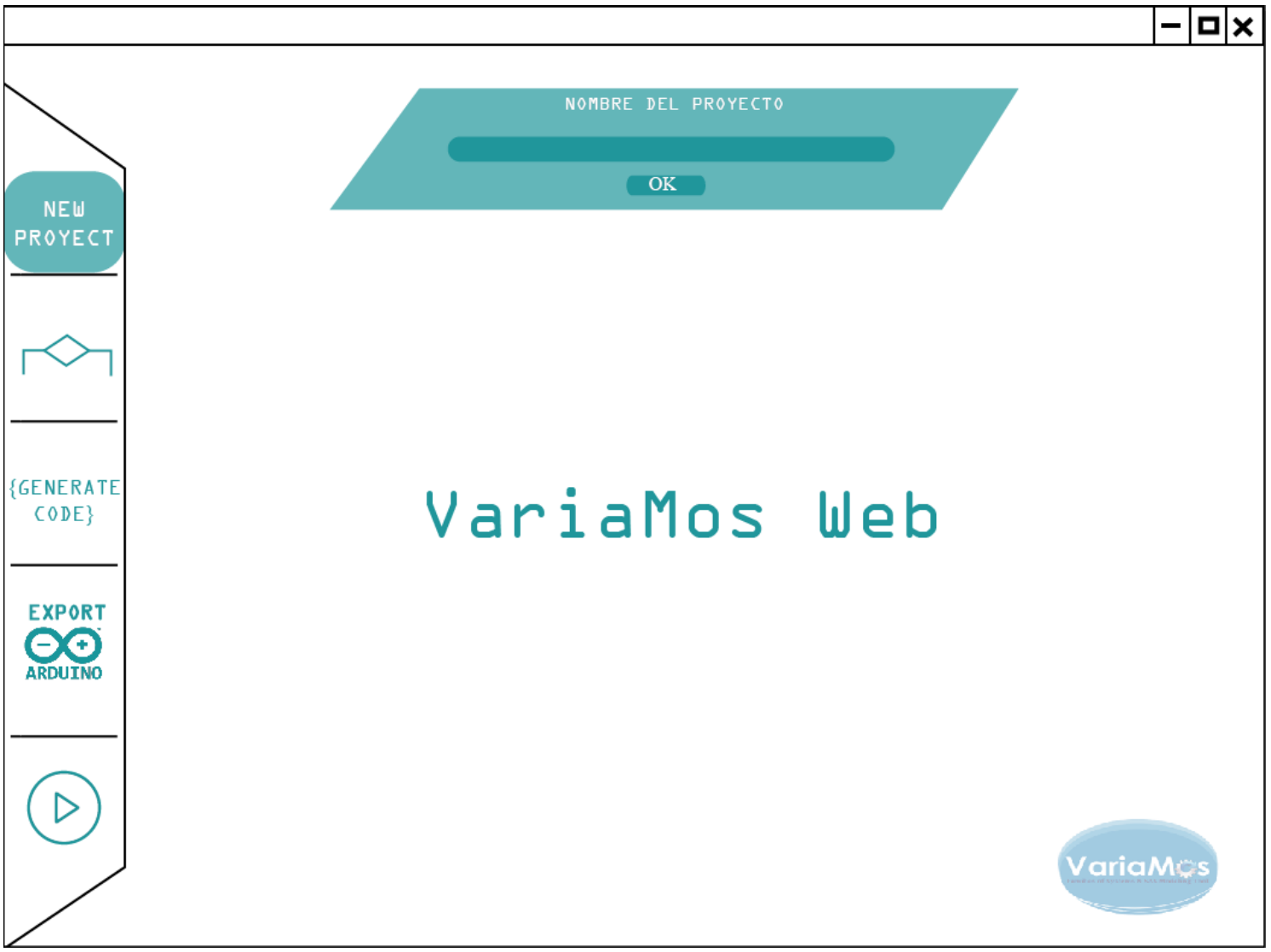
-ESTRUCTURACIÓN CORRECTA DEL DIAGRAMADO</td><td colspan="2" rowspan="4">-GENERACIÓN DE CÓDIGO CON BASE EN DIAGRAMAS

-DISEÑO INTUITIVO PARA ESTUDIANTES

-VALIDACIÓN DEL CIRCUITO</td><td>DESCRIPCIÓN DEL RIESGO</td><td>PROBABILIDAD</td><td>IMPACTO</td><td>ACCIÓN DE MITIGACIÓN</td></tr>
<tr><td>RESISTENCIA AL CAMBIO DE LOS ESTUDIANTES</td><td>MEDIA</td><td>ALTO</td><td>TUTORIALES QUE MUESTREN LAS VENTAJAS ANTE LAS PRÁCTICAS QUE TIENEN ACTUALMENTE LOS ESTUDIANTES</td></tr>
<tr><td>IDIOMA DEL USUARIO</td><td>MEDIA</td><td>MEDIO</td><td>PLATAFORMA MULTILENGUAJE</td></tr>
<tr><td>ERRORES DE PERSISTENCIA</td><td>BAJA</td><td>ALTO</td><td>RESPALDO DE PERSISTENCIA</td></tr>
</table>

2.3. Mockups


NEW
PROYECT
{GENERATE
CODE}
EXPORT
ARDUINO
VariaMos Web
VariaMos


NOMBRE DEL PROYECTO
OK
NEW PROYECT
{GENERATE CODE}
EXPORT ARDUINO
VariaMos Web
VariaMos

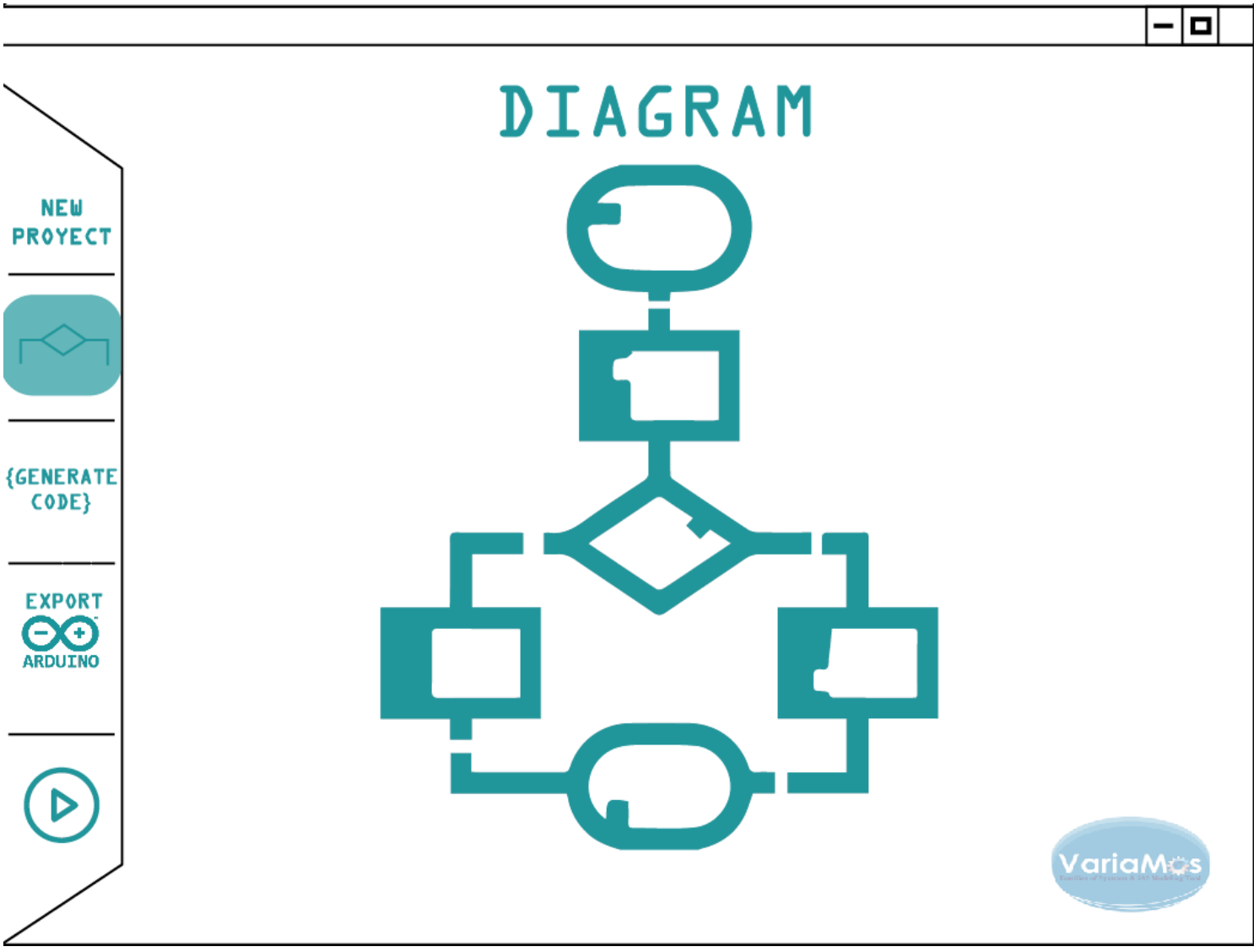
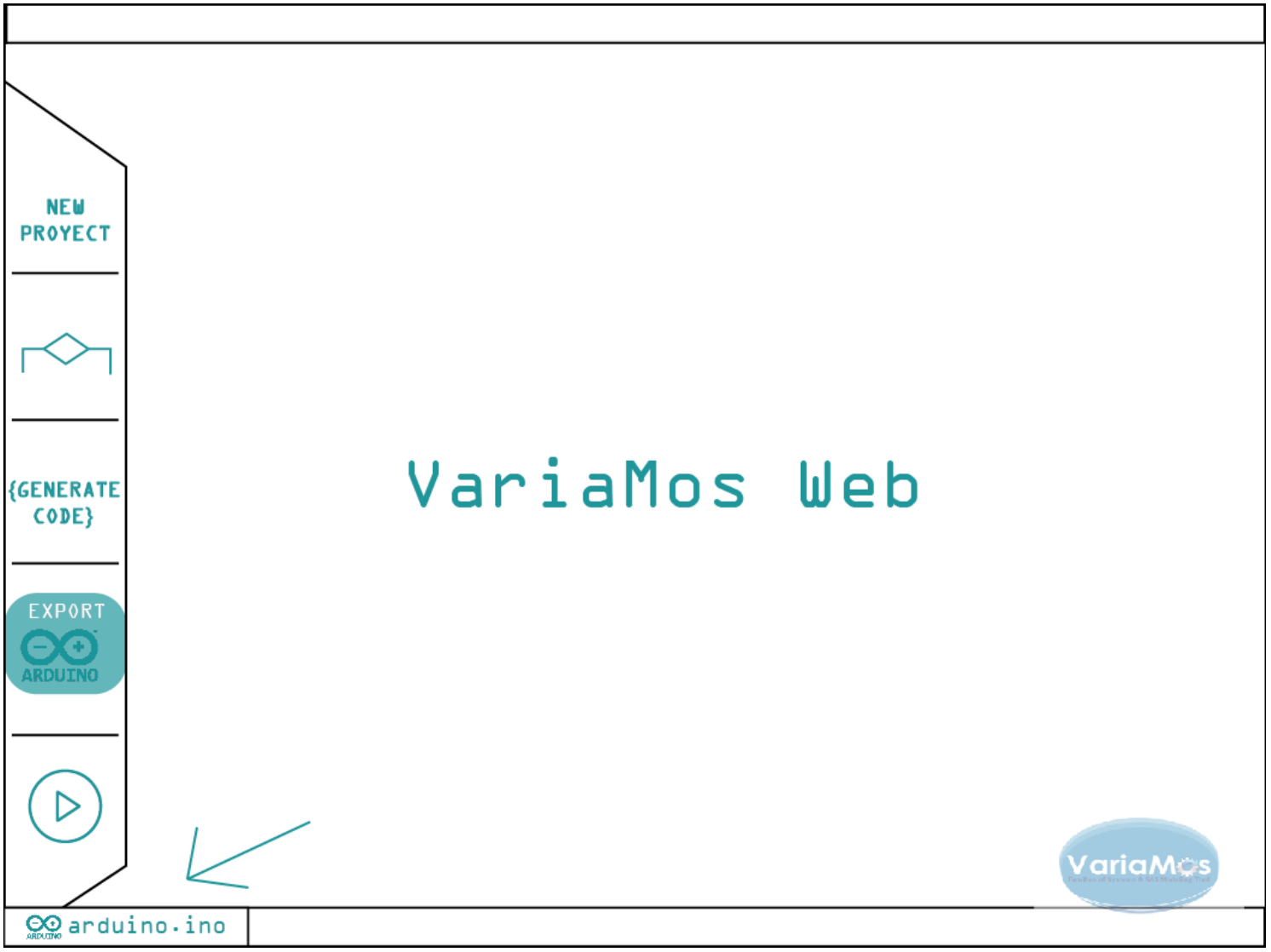
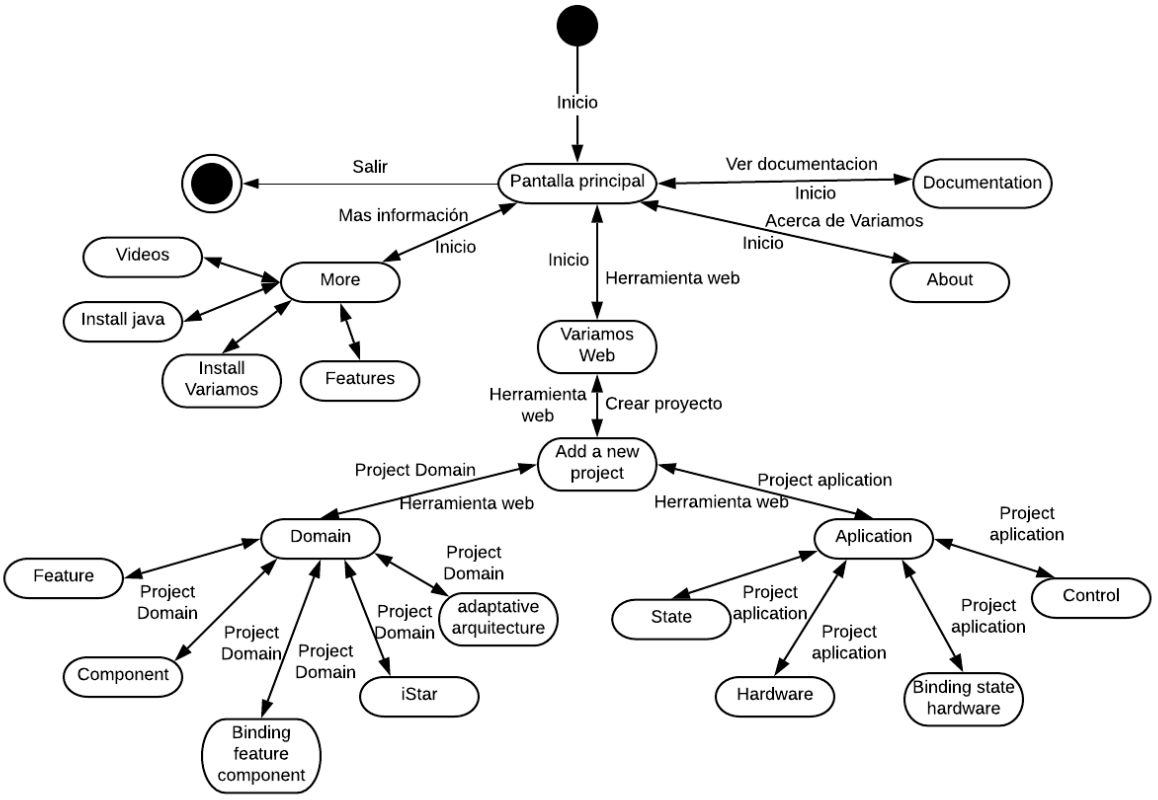


DIAGRAM
NEW PROYECT
{GENERATE CODE}
EXPORT
ARDUINO
VariaMos

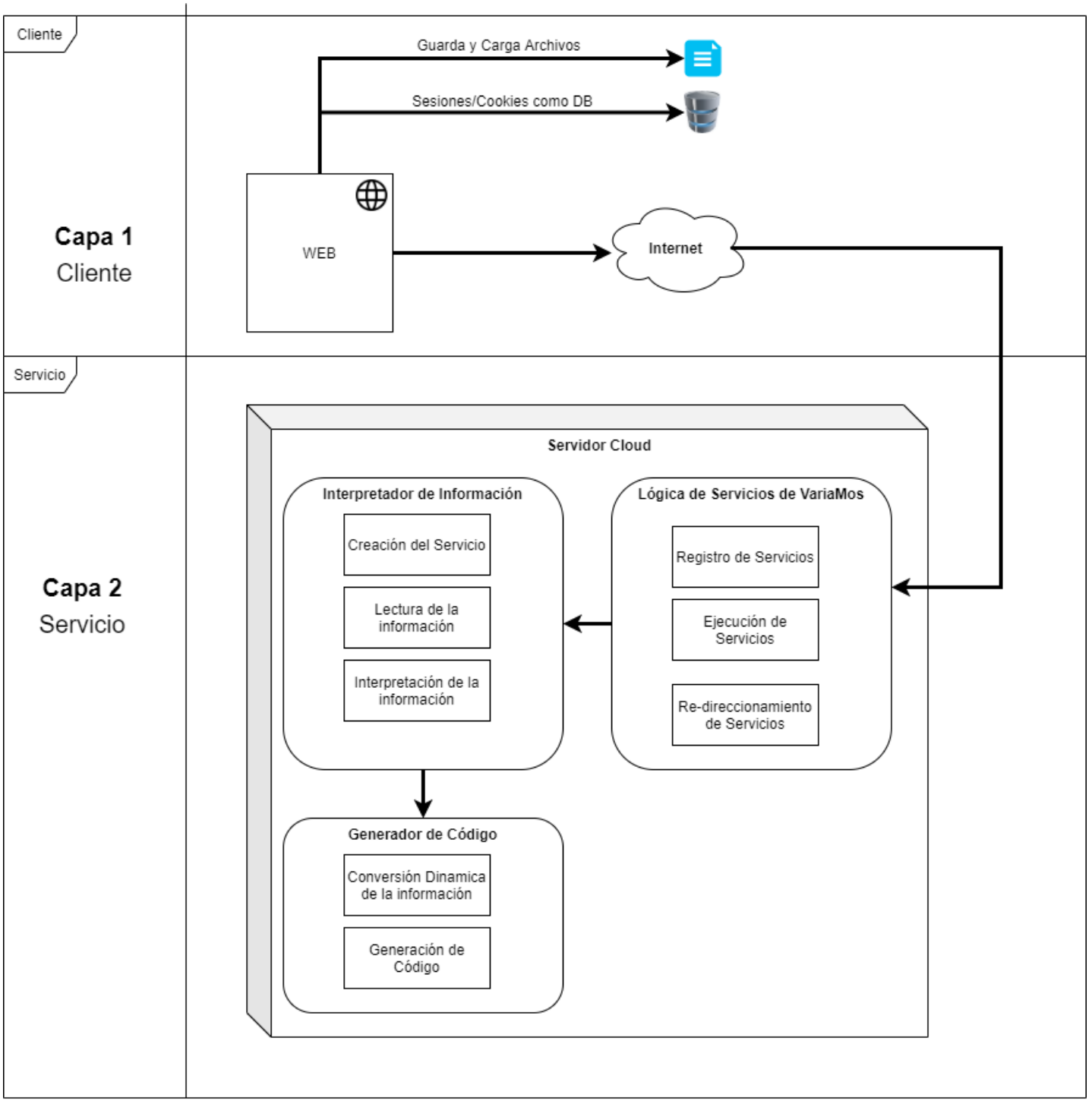
```
#define ANALOG_SENSOR_PIN A0
#define DIGITAL_SENSOR_PIN 7
#define DIGITAL_RELAY_PIN 2

int digitalValue;
int lightAnalogValue;

void setup()
{

   Serial.begin(9600);
   Serial.println("Lectura de Sensores Ejemplo 2");
   pinMode(DIGITAL_SENSOR_PIN, INPUT);
   pinMode(DIGITAL_RELAY_PIN, OUTPUT);
}

void loop(){

  digitalValue = digitalRead(DIGITAL_SENSOR_PIN);

  //Escribimos el valor de la señal
  Serial.println("Valor digital:"+ String(digitalValue));

  lightAnalogValue = analogRead(ANALOG_SENSOR_PIN);
  Serial.print("Valor analógico de (0 to 1023)");

  Serial.println(lightAnalogValue,DEC);



  if(digitalValue == LOW){
    Serial.println("Relay OFF");
    digitalWrite(DIGITAL_RELAY_PIN,LOW);
  }else{
    Serial.println("Relay ON");
    digitalWrite(DIGITAL_RELAY_PIN,HIGH);
  }

  delay(100);

}
```



2.4. Diagrama de Navegación



2.5. Diagrama de alto nivel



2.6. Matriz de solución

Capa	Componentes	Tecnologías y Frameworks
Cliente	Web	VueJS, mxGraph, HTML, CSS, NodeJS, Webpack
Servicio 1	Lógica de Servicio de Variamos	Java, Spring-Boot
Servicio 2	Interpretador de Información	Java, Gson, Lombok
Servicio 3	Generador de Código	Java

3. Requisitos del sistema

3.1. User Story Mapping

3.2. Historias de Usuario

- como desarrollador del experimento deseo que los estudiantes realicen todas sus tareas asignadas en 1 hora y 30 minutos
- como desarrollador del experimento deseo que los estudiantes realicen el poscuestionario para identificar usabilidad del sistema variamos web
- como desarrollador del experimento deseo que los estudiantes realicen el precuestionario para saber su estado antes de la prueba
- como desarrollador del experimento deseo que los estudiantes sigan el paso a paso para completar la tarea asignada
  -como desarrollador del proyecto deseo generear por parametrización de un LED RGB.
  -como desarrollador del proyecto deseo generar código en base a la parametrización de un LED y sensor DHT11.
- como desarrollador del proyecto deseo la interpretación de un JSON con información de uso por componentes electrónicos arduino.

3.3. Requisitos No Funcionales del producto

RNF-01	Desempeño de Generación
Descripción	El sistema debe entregar el código generado de manera rápida y confiable, es decir, el tiempo de respuesta debe ser muy corto.
Medida Cuantitativa	El tiempo de generación y entrega debe ser menor a 8 segundos
Importancia de la Restricción	Alta
Decisión de la Arquitectura	Un procesador apto que sea capaz de generar el código rápidamente.

RNF-02	Usabilidad
Descripción	El sistema debe tener una edición de diagramas la cual debe ser de fácil uso y entendimiento.
Medida Cuantitativa	Tiempo en minutos necesario para entender o usar funciones del sistema.
Importancia de la Restricción	Alta
Decisión de la Arquitectura	No Aplica.

RNF-03	Portabilidad
Descripción	El sistema debe poder ser utilizado en cualquier dispositivo con acceso a internet.
Medida Cuantitativa	Cantidad de sistemas operativos y dispositivos desde los que se puede acceder a la plataforma web.
Importancia de la Restricción	Media
Decisión de la Arquitectura	No Aplica.

RNF-04	Disponibilidad
Descripción	El sistema debe poder permanecer activo por periodos largos de tiempo ininterrumpidos.
Medida Cuantitativa	Tiempo en segundos que lleva activo.
Importancia de la Restricción	Alta
Decisión de la Arquitectura	Uso de un servidor Cloud que permita desplegar la plataforma durante un tiempo confiable.

RNF-05	Extensibilidad
Descripción	El sistema debe poder adaptarse a los cambios y adiciones de nuevas características.
Medida Cuantitativa	Numero de horas o días requeridos para realizar un cambio en el sistema.
Importancia de la Restricción	Media
Decisión de la Arquitectura	Uso de servicios que permitan un rápido despliegue al momento de hacer cambios.

3.4. Reuniones semanales

Las reuniones semanales fueron alrededor de 6 (la gran mayoría con el PO), sin embargo empezamos a documentarlas en la Wiki desde la tercera reunión.

Estas reuniones son indispensables puesto que trabajamos sobre un sistema ya establecido que lleva meses de desarrollo, por esta razón la comunicación ha sido constante por reuniones virtuales (su mayoría) y por medios presenciales donde vamos pactando historias de usuario que posteriormente vamos desarrollando, además de que propiciamos un entorno donde aprendemos los conceptos mas complejos de la generación del código, puesto que esto conlleva un tiempo de entendimiento de todo el código que ya había sido subido y evaluado.

Las reuniones están en la pestaña `Reuniones` de la Wiki de Moduino y están compuestas por comentarios de cada uno de los integrantes del proyecto respondiendo las preguntas de las reuniones generales con metodología SCRUM. Que son:

- ¿Que hice?
- ¿Que dificultades tuve para hacer eso?
- ¿Que haré?

4. Repositorios

En cuanto a repositorios Moduino actualmente se compone de 4 repositorios (esto podría variar en un futuro). Dichos repositorios son fundamentales y cada uno será explicado de manera breve a continuación:

- `code-generator` (Link): Este repositorio tiene todo el contenido del generador d ecódigo que se lleva trabajando desde hace casi 9 meses, dicho generador está realizado en Java y su función es hacer uso de un metamodelo como componente principal para la interpretación dinámica de objetos C/C++/Arduino Lang, como piezas de código, convirtiendo entonces objetos dinamicos de Java en código de Arduino adaptable al sistema.

- `code-binder` (Link): Este repositorio permite interpretar la información recibida desde el Front-End (variamos-web), y haciendo uso del generador (code-generator), producir el código y reenviarlo como respuesta. Se integra actualmente como un servicio al servidor de Spring-Boot (variamos-services).

- `variamos-web` (Link): Este repositorio contiene todo el desarrollo de Variamos en cuanto al Front-End, donde se utiliza como framework principal VueJS y la libreria de graficos MxGraph, para poder producir un editor de diagramas con una paleta y demás funciones y características que ofrece la plataforma a los usuarios.

- `variamos-services` (Link): Este repositorio contiene todo el desarrollo de Variamos en cuanto al Back-End, donde se utiliza Java y el framework Spring-Boot para por medio de servicios integrados manejar una API que permita procesar datos y devolver respuestas que sean complejas o que requieran de un entorno Backend. Por parte de Moduino, utilizamos el Backend para poder integrar el (code-binder) y así utilizarlo desde el FrontEnd (variamos-web).

Todos estos repositorios se encuentran en la plataforma de Github, y cada uno es parte fundamental de todo el proyecto puesto que componen el trabajo incluyendo las metas y objetivos de Moduino. Además como estamos a su vez apoyando el trabajo que ya se venía realizando desde antes, son repositorios que ya tienen un desarrollo previo y nuestra intención es permitirnos a todos los integrantes tanto de Moduino como de VariaMos un fácil acceso a nuestras copias de los repositorios originales para poder ser integrados y desplegados en la plataforma oficial (https://variamos.com/variamosweb/?#/).

4.1. Carpeta de Usabilidad

Hemos guardado los 3 documentos principales junto con los otros 2 documentos (de los experimentos en una carpeta de Google Drive), donde está toda la redacción de las pruebas que hemos diseñado, a continuación este es el link de dicha carpeta:

Link: https://drive.google.com/open?id=1DSNnWKWaDcinUMEe-i4TFOE4ZIIReZW-

5. Conclusiones

- Ahora Moduino trabaja conjuntamente con la plataforma Variamos Web.
- Moduino realizará unas pruebas de usabilidad ya documentadas y desarrolladas con la finalidad de recibir una retroalimentación real del Front-End y poder sugerir e implementar cambios.
- Moduino tuvo un avance significativo en la generación de código basado en componentes, donde se eligieron ciertos dispositivos al azar y fueron parametrizados en un JSON para poder ser interpretados con el fin de producir código adaptable.
- Las pruebas de usabilidad son una buena forma de poder conocer la opinión de los usuarios, dándoles unos experimentos que pueden realizar y luego unos cuestionarios para saber como se sintieron en cuanto a ciertos factores al realizar las pruebas.
- Es un proyecto que demuestra ser muy competitivo con el mercado actual y su objetivo de aprendizaje para estudiantes, lo cual implica en gran medida un alto impacto a los estándares y objetivos que se han planteado desde el comienzo.
- Trabajar con proyectos ya desarrollados hace parte de un análisis de cada cosa que lo compone y esto es de alta importancia ya que gracias a este análisis se puede seguir trabajando en sincronía con lo que ya se venía haciendo y se asume una responsabilidad de documentar el código para volverlo más entendible a quienes trabajarán con lo que se desarrolla.

6. Referencias Bibliográficas

[1] P. (2020). Plan before coding (why software design is needed?). [online] Howtosurviveasaprogrammer.blogspot.com. Recuperado de: http://howtosurviveasaprogrammer.blogspot.com/2018/09/plan-before-coding-why-software-design.html .

[2] Anon, (2020). [online] Recuperado de: https://www.quora.com/Which-are-the-alternatives-to-UML .

[3] VariaMos – Families of Systems & SAS Modeling Tool. (2020). Recuperado de: https://variamos.com/home/.

[4] About VariaMos – VariaMos. (2020). Recuperado de: https://variamos.com/home/about/ .

[5] Ardublock | A Graphical Programming Language for Arduino. (2020). Recuperado de: http://blog.ardublock.com/

[6] Circuit Design App for Makers- circuito.io. (2020). Recuperado de: https://www.circuito.io/

[7] Circuits on Tinkercad. (2020). Recuperado de: https://www.tinkercad.com/circuits

[8] Story Mapping, Visual Way of Building Product Backlog. (2015). Recuperado de: https://www.thoughtworks.com/insights/blog/story-mapping-visual-way-building-product-backlog

7. Vídeo

En el vídeo explicamos a detalle ciertas cosas de alta importancia y también de una manera visual mostramos el funcionamiento del sistema en base al flujo del sistema o mapa de navegación, en esta ocasión quien hace el Pitch es Daniel García.

Link del Video: https://drive.google.com/open?id=1ubf0RMxrVD9DMMs3LGf0ebS7Reja84-p

Entrega 3

Last updated by | Abraham M. Lora | 29 may. 2020 at 17:52 GMT-5

1. Introducción

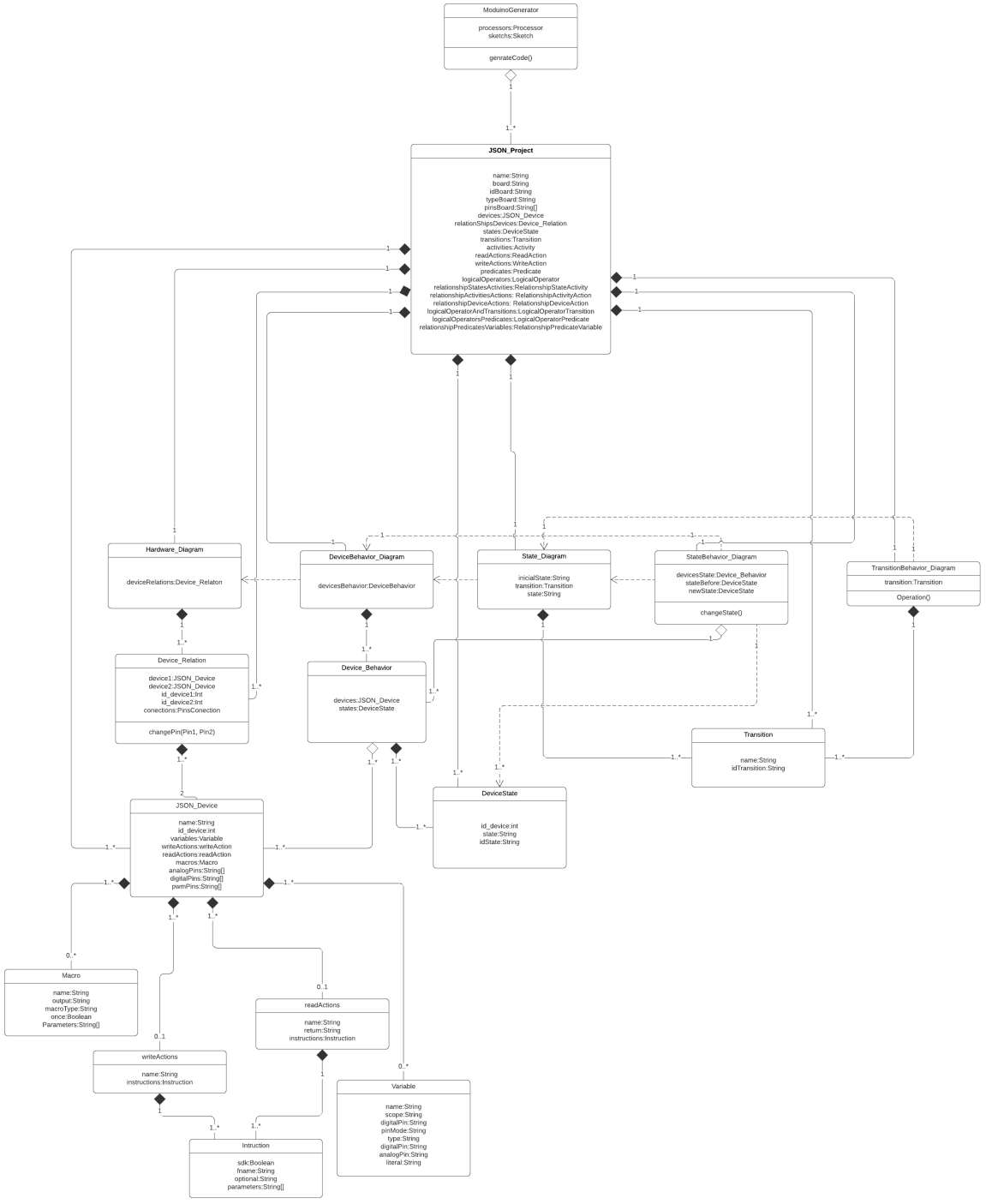
Para el presente documento se entrega el avance realizado por el equipo IoTeam tanto en la implementación de nuevos dispositivos como en la finalización de documentos para la realización de pruebas de usabilidad, ya que con esta información se tomara un registro detallado a la hora identificar dificultades en el uso de variamos Web para la generación de código Arduino; con esto se podrá implementar cambios en el front-end para mejorar su usabilidad.

Por otro lado se mostrará a través de diagramas, Mockups e imágenes una visión más completa del proyecto, de igual manera el avance realizado en cada punto, las correcciones hechas respecto a entregas anteriores y en general la evolución del proyecto que nos permite estar más cerca de lo que queremos desarrollar y lo que quiere el PO, dando solución a las necesidades y problemáticas encontradas para el ligamento y desarrollo que se estará integrando con con la plataforma de VariaMos Web para la generación de código Arduino.

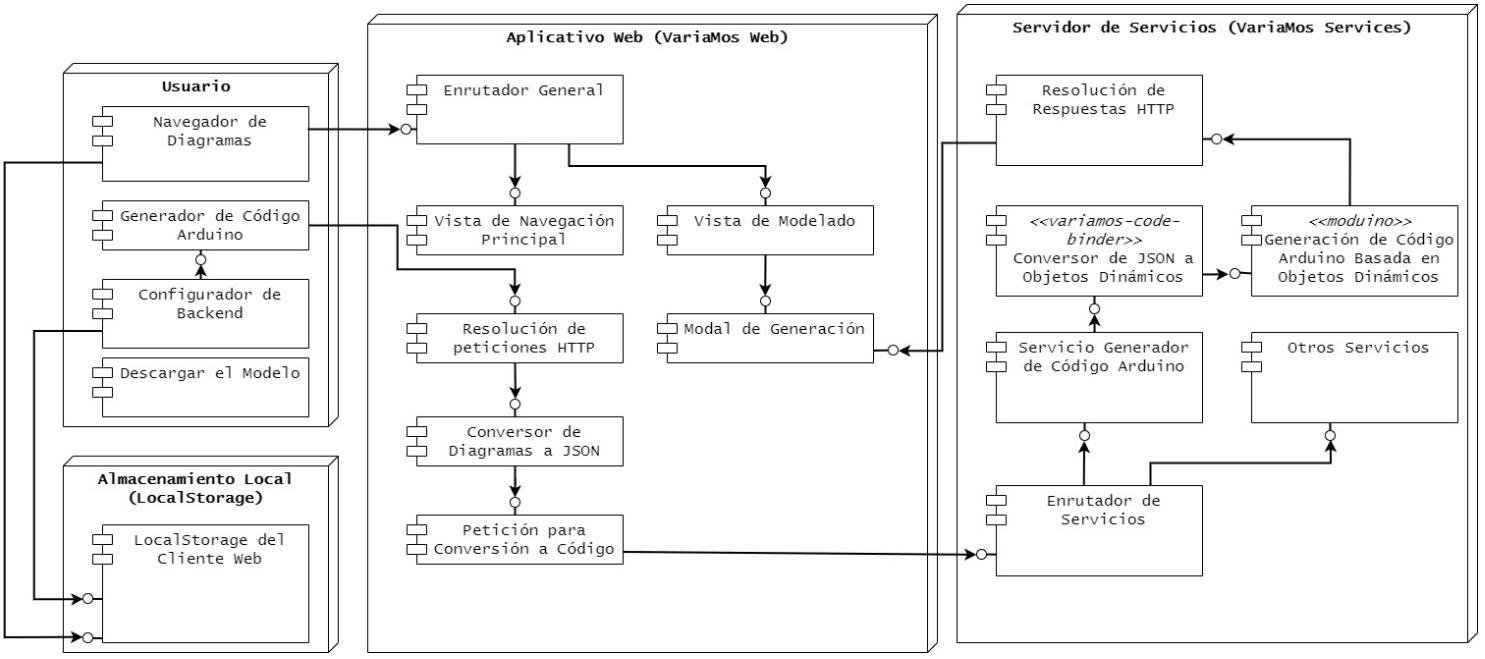
2. Arquitectura del Software

2.1 Vista Lógica

El presente diagrama de clases, representa los componentes, clases, atributos y operaciones más importantes del proyecto para su entendimiento, por lo que algunos tipos de atributos no tienen sus clases asociadas (Tipo de atributo). El generador de código representa otro proyecto de gran complejidad y por tanto en el presente diagrama se menciona como una clase, sin ser abordada con todos sus componentes y relaciones.



2.2 Vista de Implementación

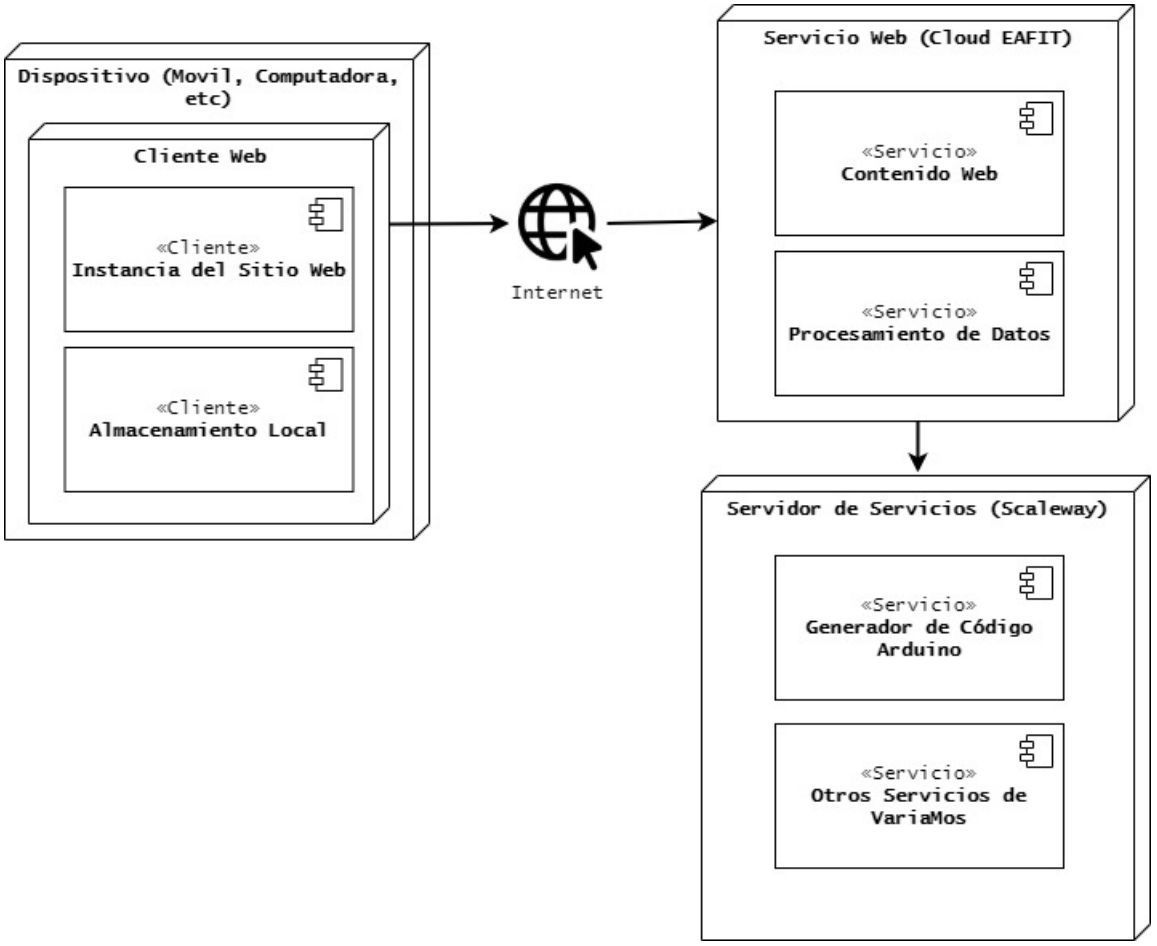


En el diagrama anterior se puede notar una división del proyecto en 4 módulos importantes, estos son:

- **Usuario:** Este modulo posee todos los componentes que hacen parte del cliente en general, siendo éstos parte fundamental de las vistas generales, ya que el aplicativo generalmente se encuentra 100% en el cliente y los datos relevantes son almacenados en el Almacenamiento Local, evitando así el uso de un backend que administre las acciones básicas del usuario.
- **Almacenamiento Local:** Este modulo hace referencia total al almacenamiento local de cada Usuario, basándose entonces en un almacenamiento meramente del cliente que utilice el usuario, ya que el aplicativo en su totalidad (exceptuando los servicios), son almacenados aquí.
- **Aplicativo Web:** Este modulo aunque sigue haciendo parte del cliente, implica toda la funcionalidad lógica que se procesa en base a las acciones del cliente durante las vistas. Se compone entonces de interacciones con el usuario y de procesamientos de datos y diagramas, encargándose también del envío de peticiones y comunicaciones con el servidor de servicios.
- **Servidor de Servicios:** El servidor de Servicios, también conocido como "*VariaMos Services*", es el backend encargado de dar funcionalidad los sistemas donde la carga en el cliente puede ser muy compleja y disminuir el rendimiento de la plataforma, por tanto, los servicios procesan esto en un servidor aparte y la información es devuelta al cliente dándole la oportunidad de obtener los resultados de su diagramado.

> En base a estos modulos se realiza todo el proceso del componente de diagramado de proyectos Arduino incluyendo el proceso de generación de código y las etapas.

2.3 Vista de Despliegue del Producto

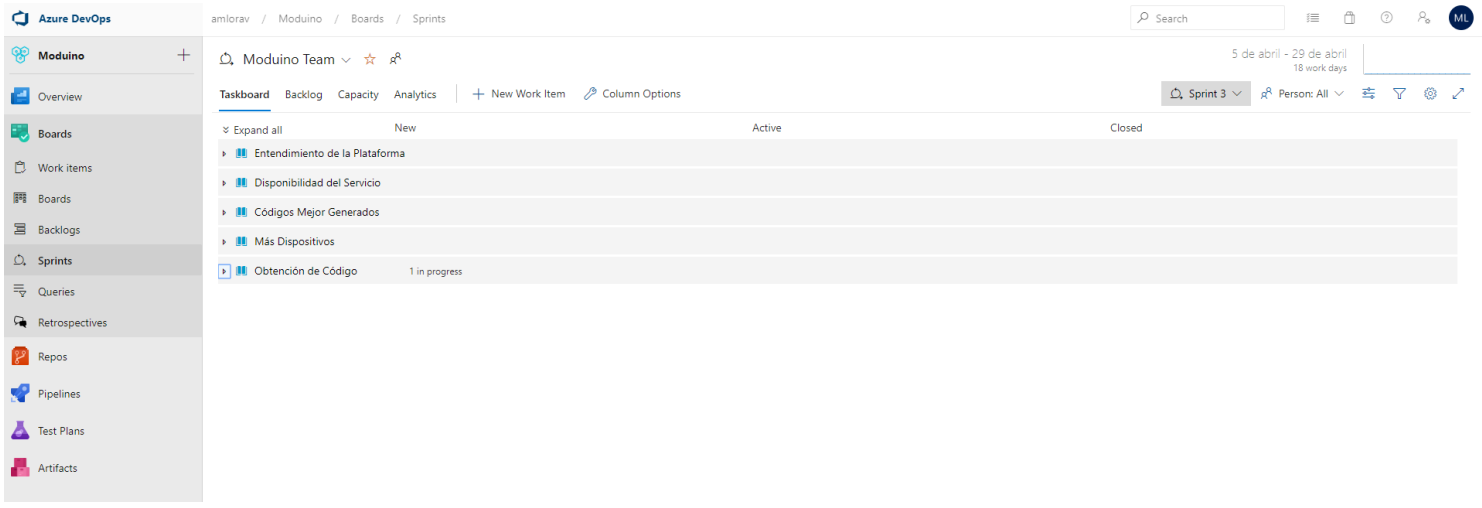


En el diagrama de despliegue se dan los 3 elementos importantes que permiten la funcionalidad completa de la plataforma, en este caso se tiene el Cliente, el cual es el punto de partida y representa en sí al usuario, su dispositivo y su navegador, posteriormente se encuentra el Servicio Web el cual se despliega en un servidor (en este caso está siendo desplegado en los servidores de EAFIT), y finalmente el servidor de servicios, que está siendo desplegado por nosotros en una plataforma llamada Scaleway.

Links Últiles:

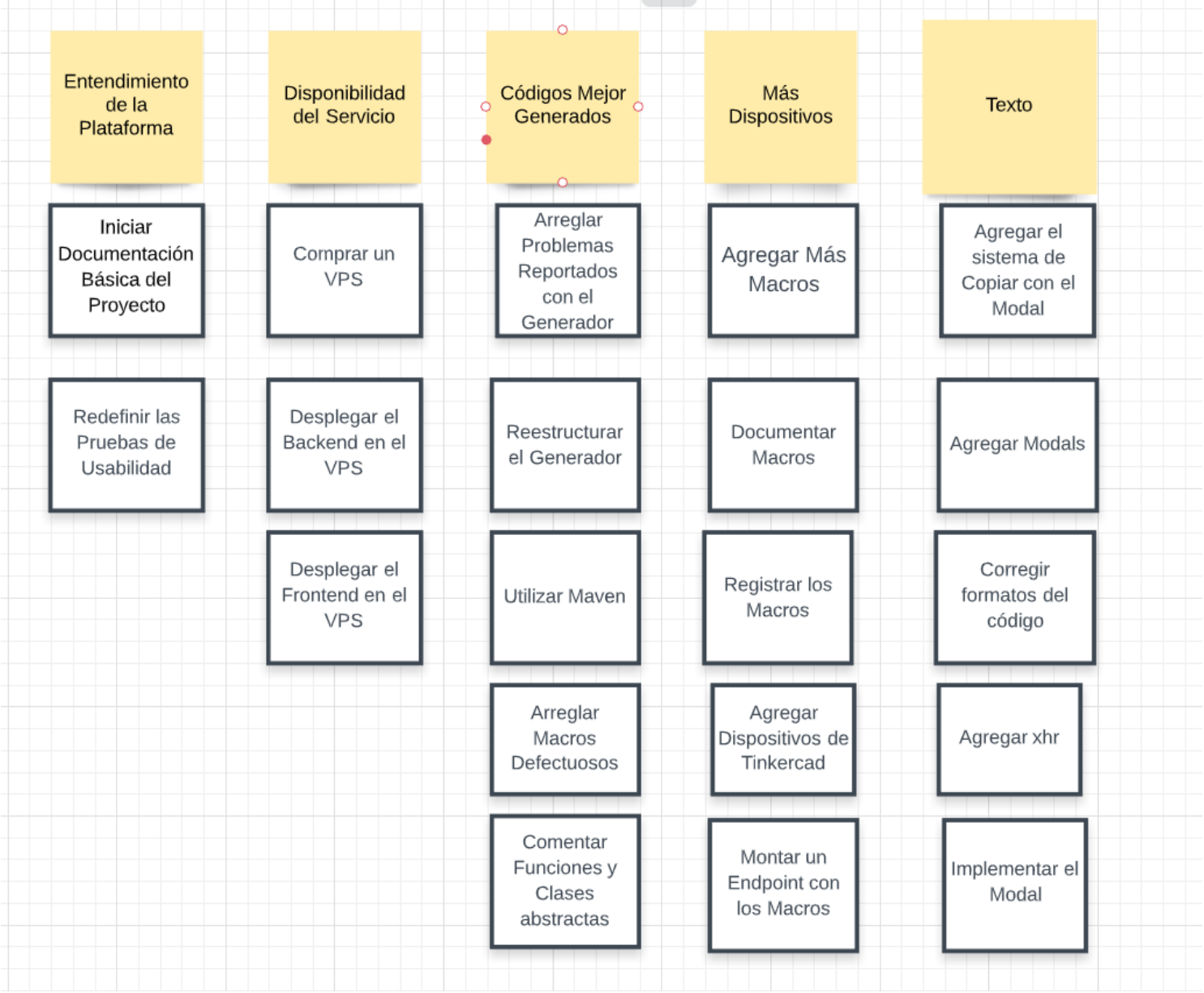
- Servicio Web: http://variamos.dis.eafit.edu.co/
- Servidor Backend: http://51.15.209.191:8080/
- Scaleway Cloud: https://www.scaleway.com/

3. Historias de usuario
En el siguiente link estaran ubicado las historias de usuario:
https://dev.azure.com/amlorav/Moduino/_sprints/taskboard/Moduino Team/Moduino/Sprint 3



4. Story Mapping

https://dev.azure.com/amlorav/Moduino/_sprints/taskboard/Moduino Team/Moduino/Sprint 3



5. Reuniones Semanales

Las reuniones semanales fueron alrededor de 4 (la gran mayoría con el PO y Paola), sin embargo empezamos a documentarlas en la Wiki la ultima semana.

Estas reuniones se han hablado temas importantes tanto de implementaciones, usabilidad y cambios en el desarrollo de VariaMos Web con el P.O.
Esta comunicación ha sido constante por reuniones virtuales.

Las reuniones están en la pestaña Reuniones de la Wiki de Moduino y están compuestas por comentarios de cada uno de los integrantes del proyecto respondiendo las preguntas de las reuniones generales con metodología SCRUM. Que son:

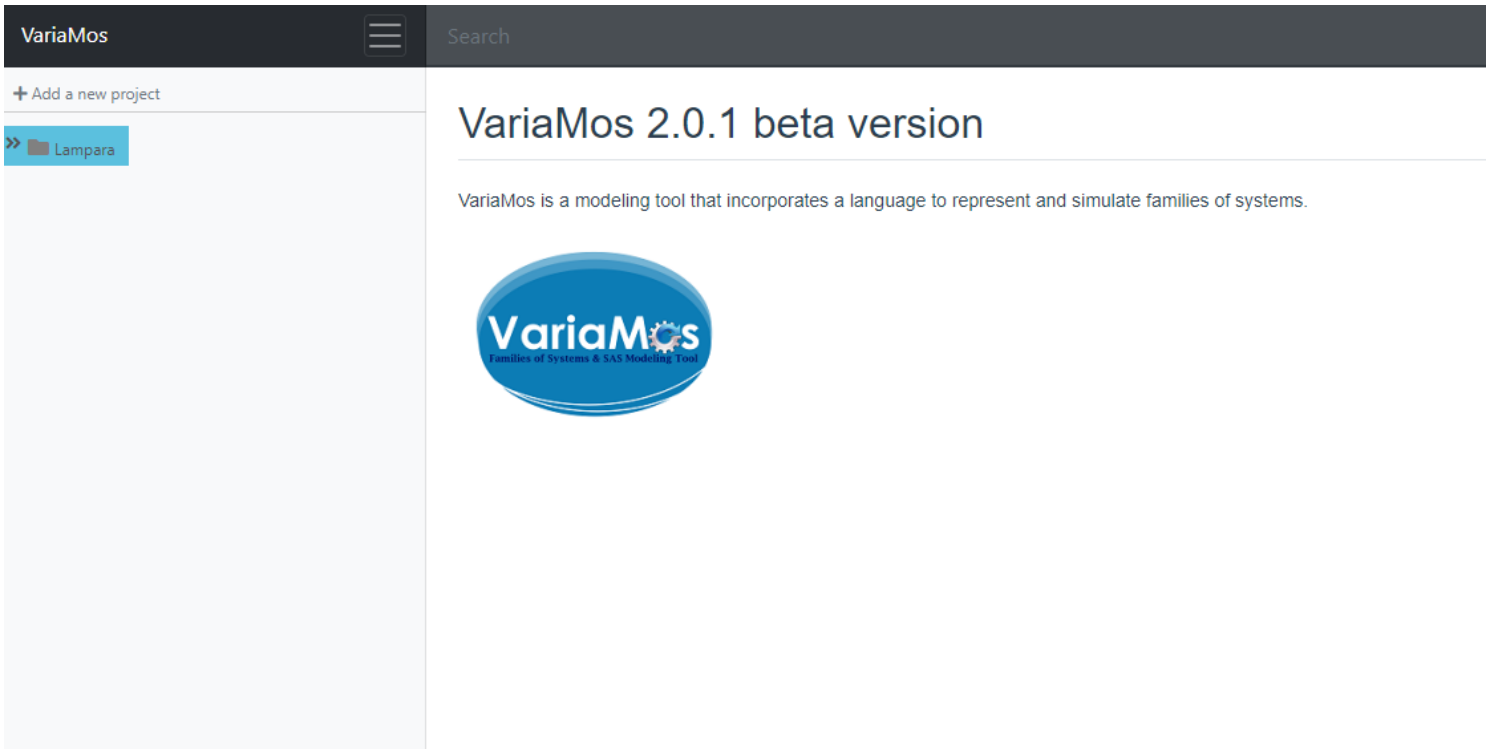
¿Que hice?
¿Que dificultades tuve para hacer eso?
¿Que haré?

A continuación se adjunta el link directo de las reuniones
https://dev.azure.com/amlorav/Moduino/_wiki/wikis/Unknow-Project.wiki/7/Reuniones-Semanales

6. Mockups

Cabe mencionar que nuestros cambios han sido màs en el back-end ya que sin haber realizado las pruebas de usabilidad no se puede realizar cambios notorios en el Frond-end

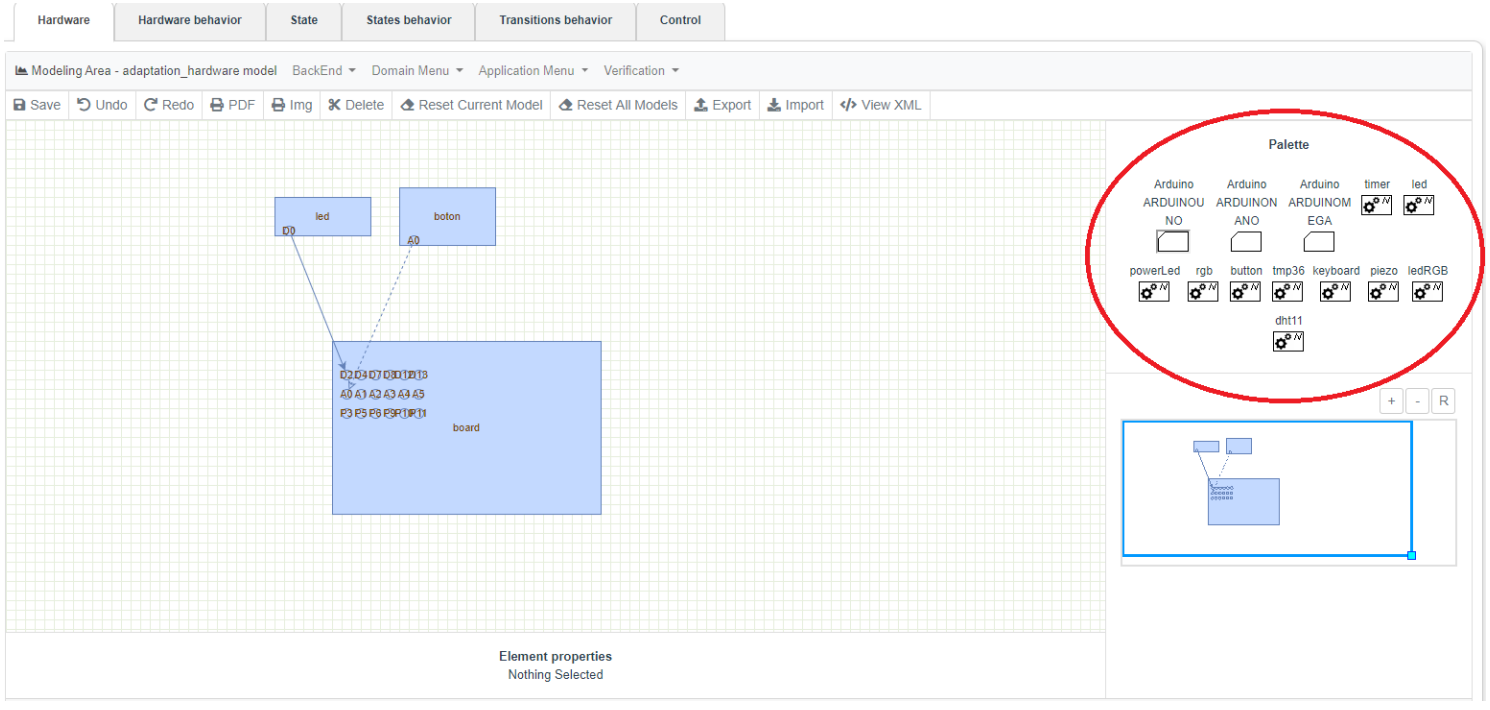
Inicio de variamos Web (**Realizado por VairaMos**)



Pestaña de modelado de hardware

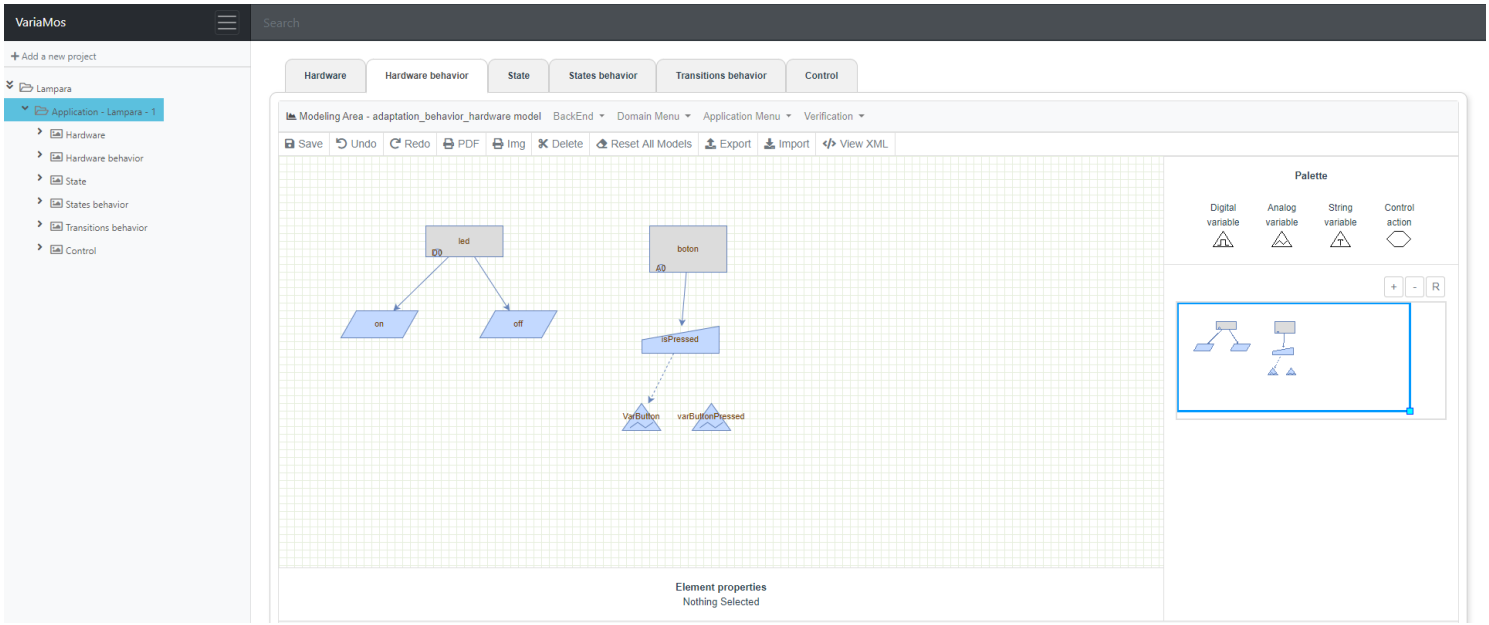
Con este modelo se representan los componentes físicos del sistemas.

Lo encerrado en un circulo rojo fue diseñado con el equipo IoTeam el cual son los componentes de hardware



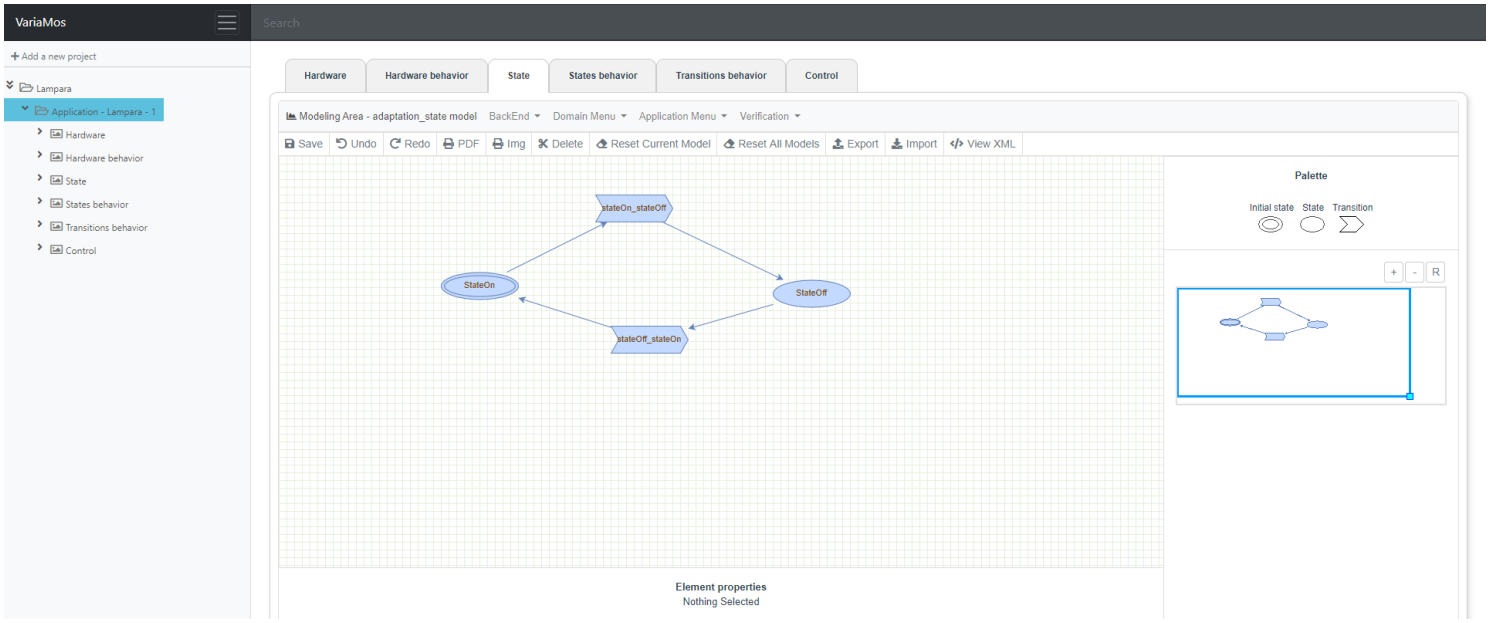
pestaña de comportamiento de hardware (**Realizado por VairaMos**)

Con este modelo se representan los procedimientos y las funciones que serán ejecutadas por cada dispositivo físico del sistema.



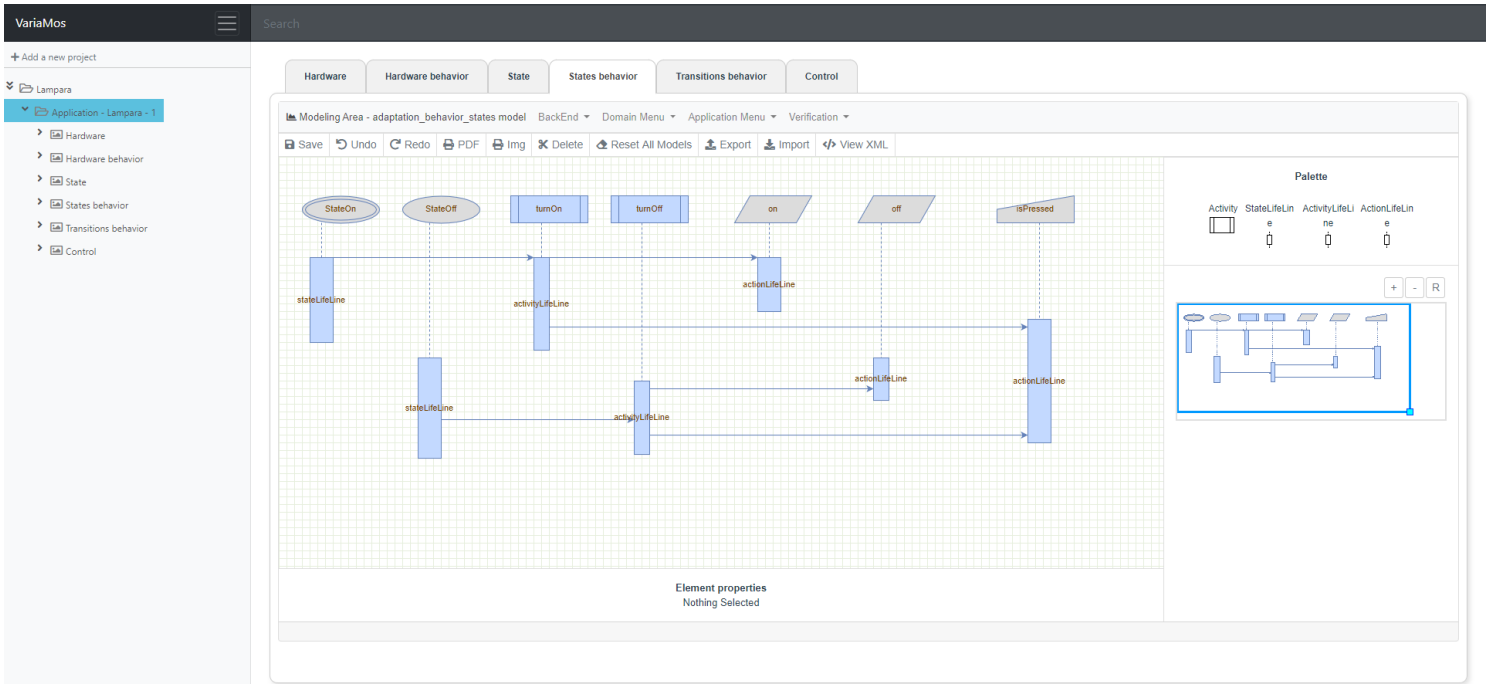
Pestaña de modelado de estado

Con este modelo se representan los estados en los que se puede encontrar el sistema y las transiciones entre ellos.



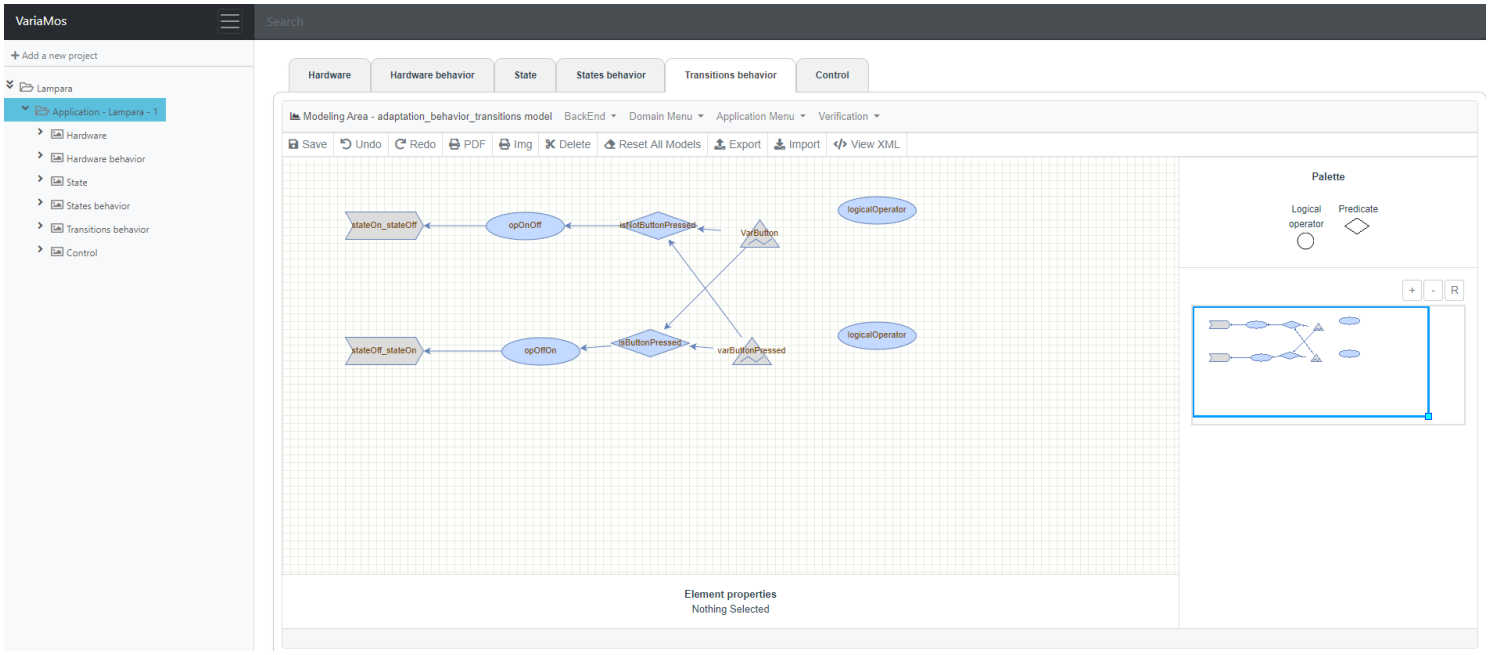
modelado de comportamiento de estado (**Realizado por VairaMos**)

Con este modelo se representa la invocación de los procedimientos y funciones de los dispositivos físicos en cada uno de los estados.

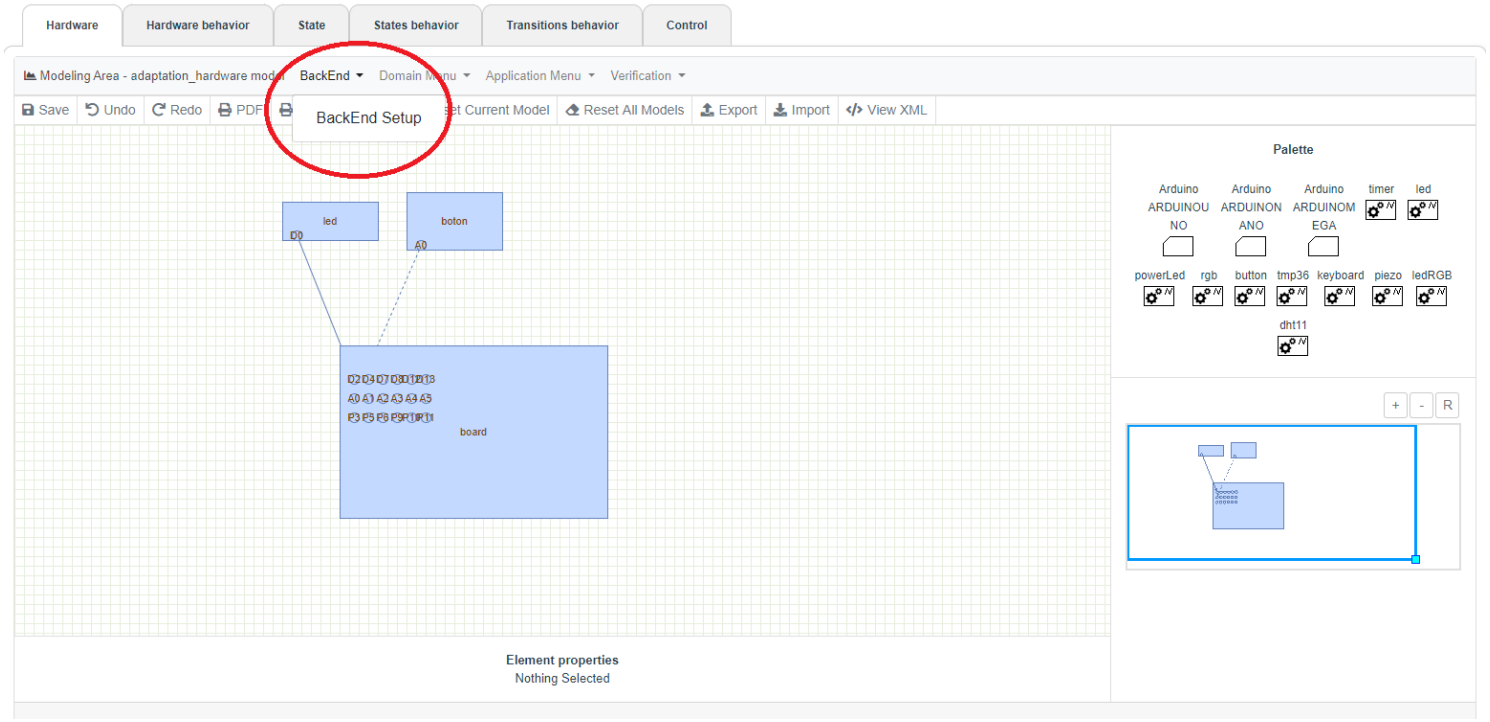


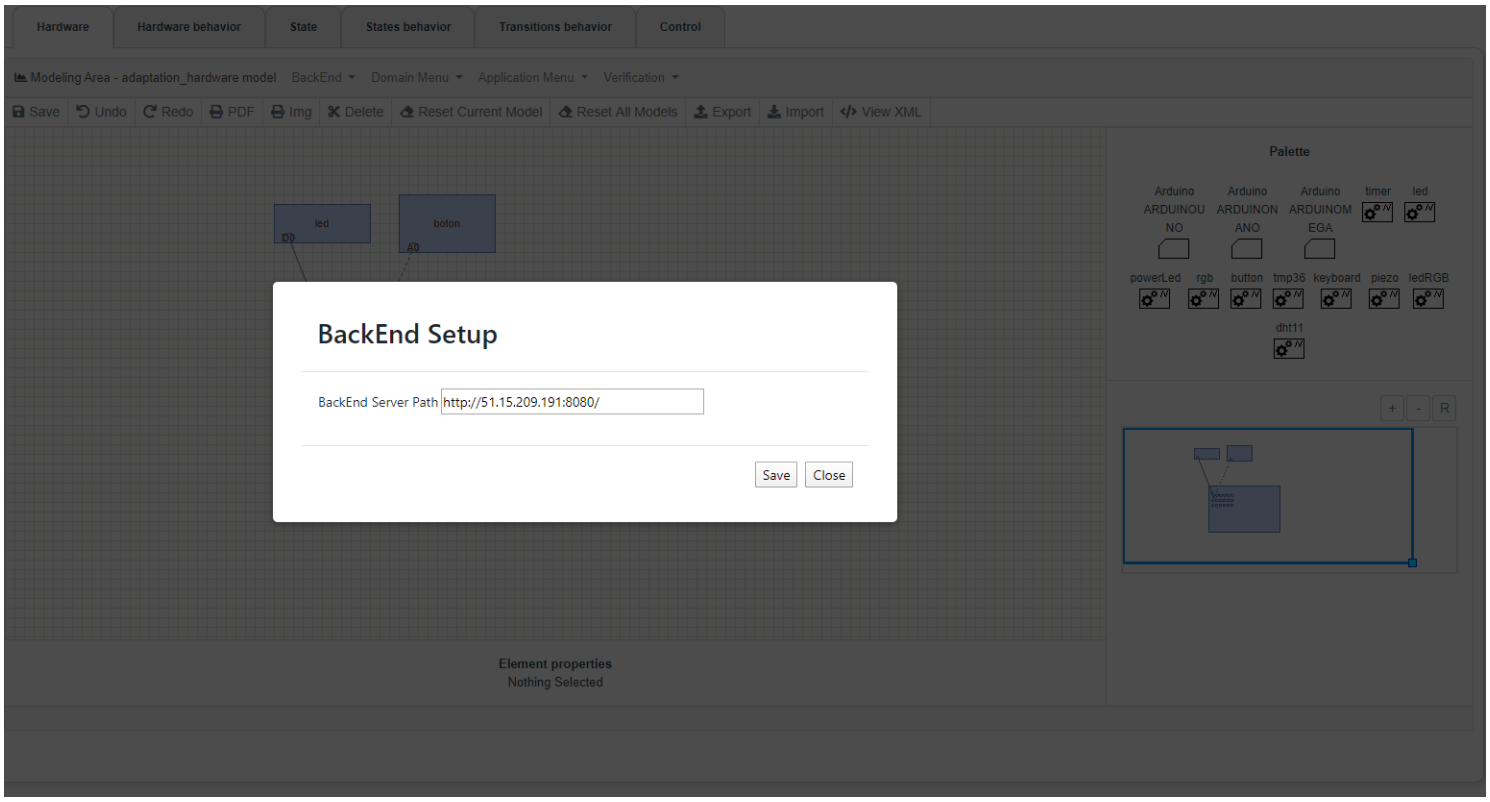
Modelo de comportamiento de transiciones(**Realizado por VairaMos**)

Con este modelo se representan las condiciones que se deben cumplir para la transición entre estados.

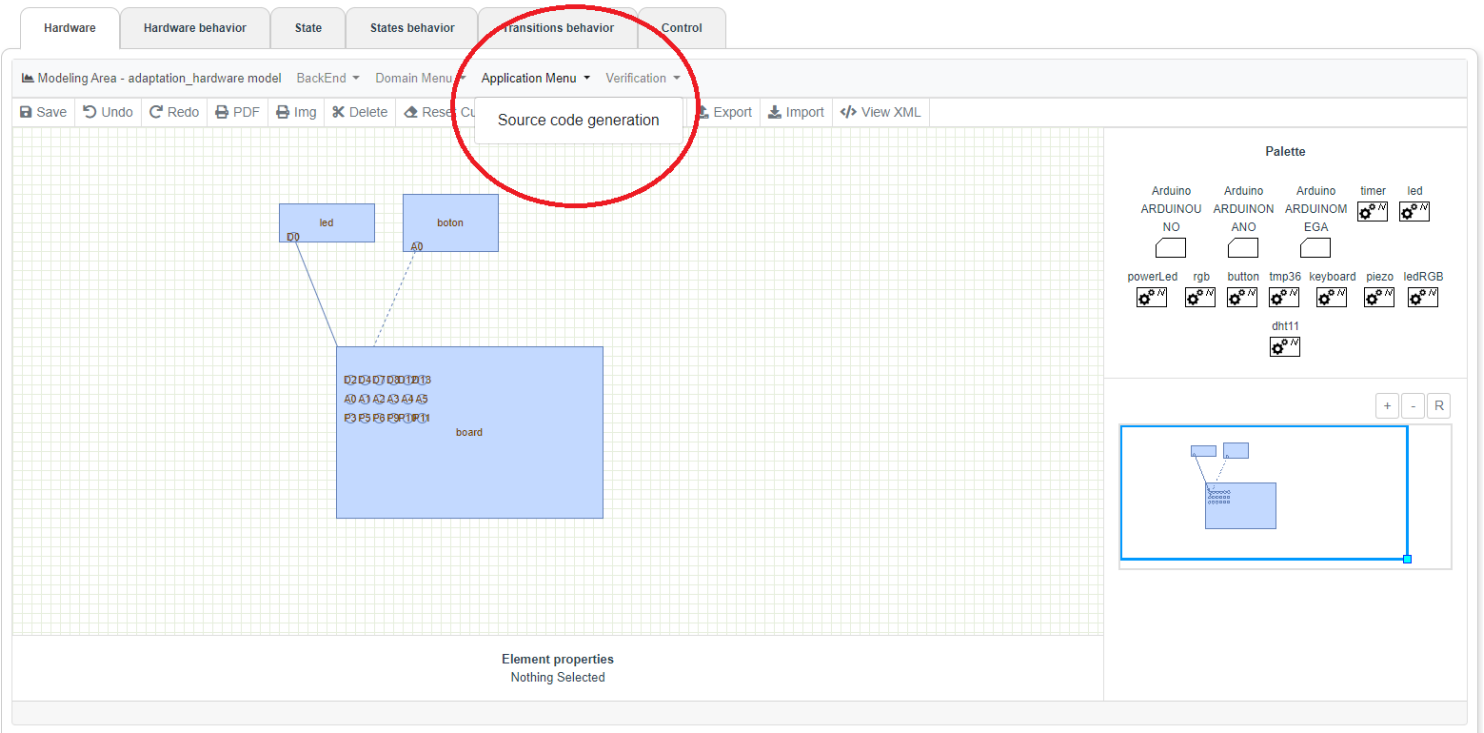


Lo encerrado en un circulo rojo fue diseñado con el equipo IoTeam el cual el BackEnd Setup que sirve para conectar el Frond-End con el Back-End

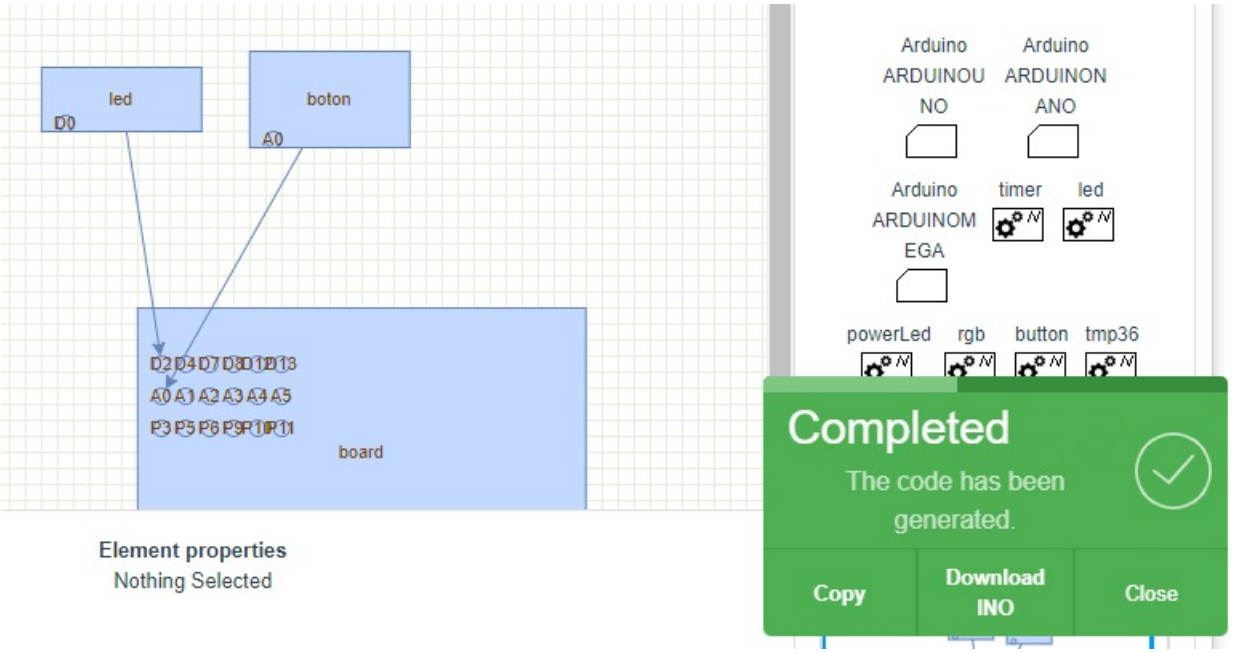




Lo encerrado en un circulo rojo fue diseñado con el equipo IoTeam el cual el Source code generation que sirve para generar el codigo una vez ya este hecho el modelado de todos sus componentes



Lo encerrado en un circulo rojo fue diseñado con el equipo IoTeam el cual sirve para decirte si el código se genero con éxito o con error, ademas se permite la copia del código directamente o la descarga del archivo Ardiuno.



7. Repositorio

Actualmente Moduino se mantiene igual que antes, debido a que en cuanto a repositorios actualmente el proyecto se compone de 4 repositorios. Dichos repositorios son fundamentales y cada uno será explicado de manera breve a continuación:

- `moduino` (Link): Este repositorio tiene todo el contenido del generador de código que se lleva trabajando desde hace casi 10 meses, dicho generador está realizado en Java y su función es hacer uso de un metamodelo como componente principal para la interpretación dinámica de objetos C/C++/Arduino Lang, como piezas de código, convirtiendo entonces objetos dinamicos de Java en código de Arduino adaptable al sistema, el cambio general con la anterior versión es que se ha refactorizado por completo a petición de VariaMos por una implementación en Maven del sistema, por tanto el repositorio ahora no se llamad `code-generator` si no que pasa a llamarse `moduino` el cual es el nombre de nuestro proyecto integrador, permitiendo una mejor visibilidad y entendimiento de nuestro desarrollo y aporte a VariaMos.

- `variamos-code-binder` (Link): Este repositorio permite interpretar la información recibida desde el Front-End (variamos-web), y haciendo uso del generador (moduino), producir el código y reenviarlo como respuesta. Se integra actualmente como un servicio al servidor de Spring-Boot (variamos-services). A su vez, este repositorio también fue cambiado de nombre, permitiendo un mejor entendimiento y se realizaron refactorizaciones en clases y presentación de los repositorios.

- `variamos-web` (Link): Este repositorio contiene todo el desarrollo de Variamos en cuanto al Front-End, donde se utiliza como framework principal VueJS y la libreria de graficos MxGraph, para poder producir un editor de diagramas con una paleta y demás funciones y características que ofrece la plataforma a los usuarios.

- `variamos-services` (Link): Este repositorio contiene todo el desarrollo de Variamos en cuanto al Back-End, donde se utiliza Java y el framework Spring-Boot para por medio de servicios integrados manejar una API que permita procesar datos y devolver respuestas que sean complejas o que requieran de un entorno Backend. Por parte de Moduino, utilizamos el Backend para poder integrar el (code-binder) y así utilizarlo desde el FrontEnd (variamos-web).

Todos estos repositorios se encuentran en la plataforma de Github, y cada uno es parte fundamental de todo el proyecto puesto que componen el trabajo incluyendo las metas y objetivos de Moduino. Además como estamos a su vez apoyando el trabajo que ya se venía realizando desde antes, son repositorios que ya tienen un desarrollo previo y nuestra intención es permitirnos a todos los integrantes tanto de Moduino como de VariaMos un fácil acceso a nuestras copias de los repositorios originales para poder ser integrados y desplegados en la plataforma oficial (http://variamos.dis.eafit.edu.co/#/).

8. Conclusiones

Durante el desarrollo de este Sprint 3, hemos realizado bastantes cambios estructurales, tanto a los repositorios donde subimos y compartimos todos los cambios de desarrollo, así como también las dificultades que hemos tenido durante la ejecución preliminar de las pruebas de Usabilidad. Sin embargo, muchos de los cambios que hemos realizado son parte de la estructura base del proyecto y gracias a estos mejoramos y refinamos poco a poco el desarrollo de la plataforma, haciéndola más amigable para el usuario y corrigiendo errores básicos que empeoraban la experiencia. Por tanto, consideramos que ha sido un Sprint muy enfocado al desarrollo base y lógico de la plataforma y que gracias a estas mejoras nos permitirá un mejor y amplío desarrollo en las pruebas de usabilidad para así poder cumplir el objetivo deseado.

9. Referencias

[1] New Bare Metal Offer (n.d.). [online] Recuperado de: https://www.scaleway.com/en/ .

[2] Anon, (2020). [online] Recuperado de: https://www.quora.com/Which-are-the-alternatives-to-UML .

[3] VariaMos – Families of Systems & SAS Modeling Tool. (2020). Recuperado de: https://variamos.com/home/ .

[4] Story Mapping, Visual Way of Building Product Backlog. (2015). Recuperado de: https://www.thoughtworks.com/insights/blog/story-mapping-visual-way-building-product-backlog

[5] IBM Knowledge Center. (2014). Recuperado de: https://www.ibm.com/support/knowledgecenter/SS8PJ7_9.5.0/com.ibm.xtools.modeler.doc/topics/cdepd.html

[6] UML Deployment Diagram. (2016). Recuperado de: https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-deployment-diagram/

10. Entregable en video
https://www.youtube.com/watch?v=9X3h5a9xRN8&feature=youtu.be

Entrega 4

Last updated by | Mateo Montes Loaiza | 2 jun. 2020 at 22:46 GMT-5

1. Problemática

Para nadie es un secreto que la programación y el modelado son elementos de gran importancia para los estudiantes de ingeniería de la universidad EAFIT, ya que estos conocimientos son los que lo llevan al diseño y desarrollo de soluciones innovadoras, que aporten a la construcción social y lo posicionen en la industria. En este proceso de aprendizaje universitario, gran parte de los estudiantes, tiene pocos o nulos conocimientos tanto en programación como en el diseño y modelado de soluciones, de manera que durante su proceso incurren gran parte del tiempo en métodos de prueba y error, o aprenden directamente en la práctica, de igual manera los diseños pedidos por los profesores son olvidados rápidamente y quedan inútiles a la hora del desarrollo. Como consecuencia de las fallas mencionadas, los estudiantes tardan mucho más tiempo en dar soluciones, por lo que muchas materias, no logran avanzar tanto cómo se esperaría, o los estudiantes desarrollan sus tareas sin adquirir estos importantes conocimientos de modelado y programación.

La problemática planteada está sustentada en los sentimientos expresados por muchos estudiantes de ingeniería y por palabras de los profesores que dictan este tipo de materias como electrónica digital y teoría de la conmutación, todo lo anterior se documentó y se llevo a cabo mediante entrevistas, encuestas y grabaciones.

2. Propuesta de solución

Desde IoTeam, planteamos una propuesta de solución a la que llamamos "Moduino", esta propuesta consiste en el mejoramiento de la plataforma del modulo de sistemas embebidos llamada "VariaMos web", una plataforma que utiliza la creación de diagramas básicos que modelan el hardware y el comportamiento de este para generar código en circuitos electrónicos que utilizan herramientas como Arduino. La plataforma no era 100% funcional en la generación de código, ni tenía una estructura para representara el comportamiento de los dispositivos, por lo que nuestro equipo trabajó conjuntamente, para lograr que esta plataforma generara código 100% funcional, que entendiera el comportamiento de los dispositivos y lo transmitiera directamente a el código. Nuestra propuesta entonces, permite al estudiante enfocar sus conocimientos en el diseño y modelado previo al desarrollo, de manera que se preocupa primero por diseñar una solución innovadora que posteriormente podrá plasmar en los diagramas logrando una visión total del sistema que le generará código automáticamente para Arduino.

Actualmente, existen en la industria, herramientas como Proteus, Scratch, Tinkercad y similares, que modelan circuitos y permiten la generación de un código base, pero no utilizan los elementos del diagramado que permiten representar estados y comportamientos de los dispositivos, logrando entonces con nuestra propuesta, una solución innovadora que habilita al usuario modelar tanto estados como comportamientos y generar código funcional, donde los dispositivos pasan de un estado a otro con el comportamiento especificado por el usuario.

3. Imagen Corporativa

Para el desarrollo de este punto se debe tener en cuenta que VariaMos actualmente está en busca de una imagen corporativa, y para ello aceptan propuestas por parte de los participes del proyecto. Por parte de Moduino hemos desarrollado un logo y lema, y hemos realizado algunos montajes de como se vería el logo si oficialmente fuese parte de la plataforma en estos momentos, estos han sido los resultados:

Colores:

- Principal: `#008ed2`
- Secundario: `#aadbee`

Logo:

- Esta propuesta de logo la hemos realizado basandonos rotundamente en los colores principales que ya son utilizados actualmente por VariaMos. Como Isotipo hemos utilizado una letra V la cual posee 3 pliegues como si se tratase de un libro, donde el significado que le hemos atribuido es que representa la variedad de diagramados que se pueden realizar en la plataforma.

Lema: *¡Modelando ideas!*

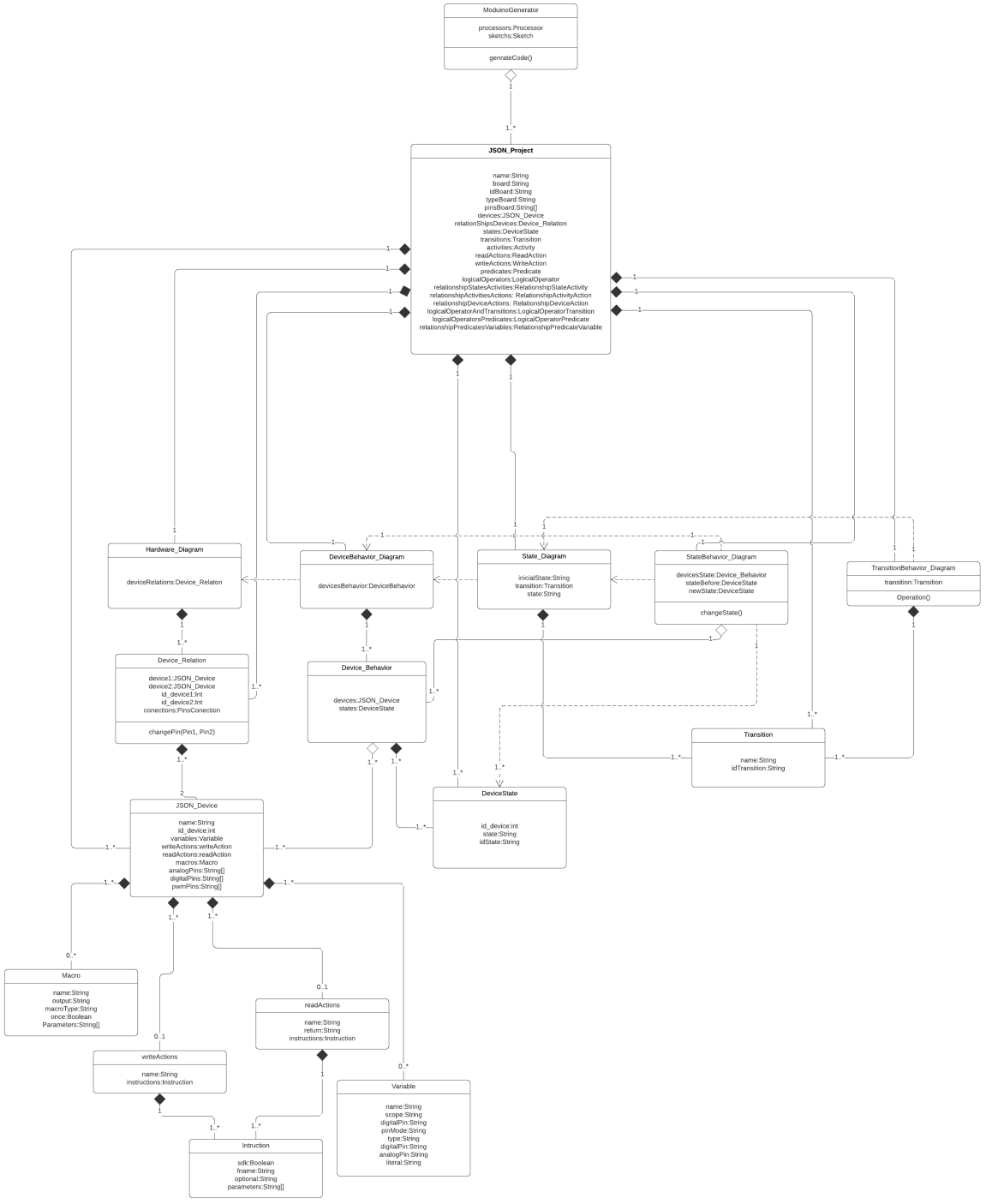
Capturas de las Interfaces:

Calidad completa de Logotipos e Isotipos (drive): https://drive.google.com/drive/folders/1YqNTSWSYPmVxyWpi1k7ey4Cim2zStQd2?usp=sharing

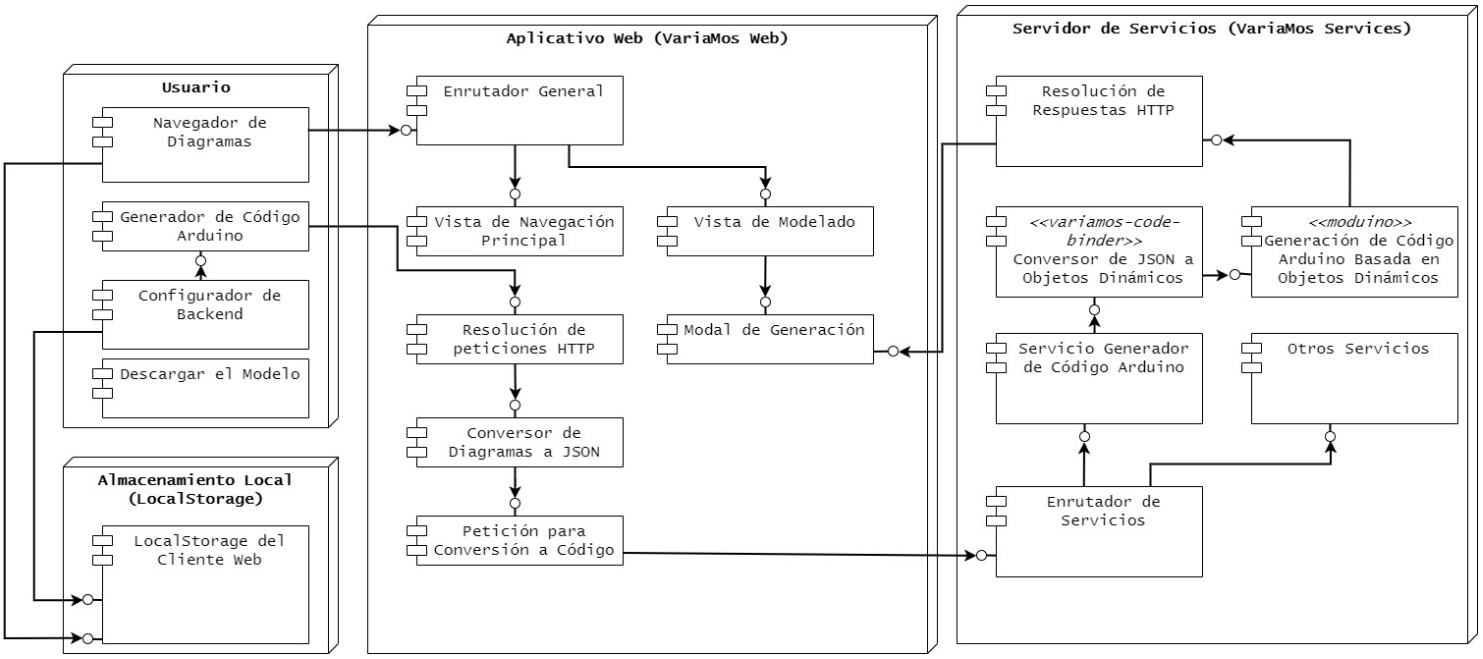
4. Arquitectura del Software

4.1. Vista Lógica

El presente diagrama de clases, representa los componentes, clases, atributos y operaciones más importantes del proyecto para su entendimiento, por lo que algunos tipos de atributos no tienen sus clases asociadas (Tipo de atributo). El generador de código representa otro proyecto de gran complejidad y por tanto en el presente diagrama se menciona como una clase, sin ser abordada con todos sus componentes y relaciones.



4.2. Vista de Implementación

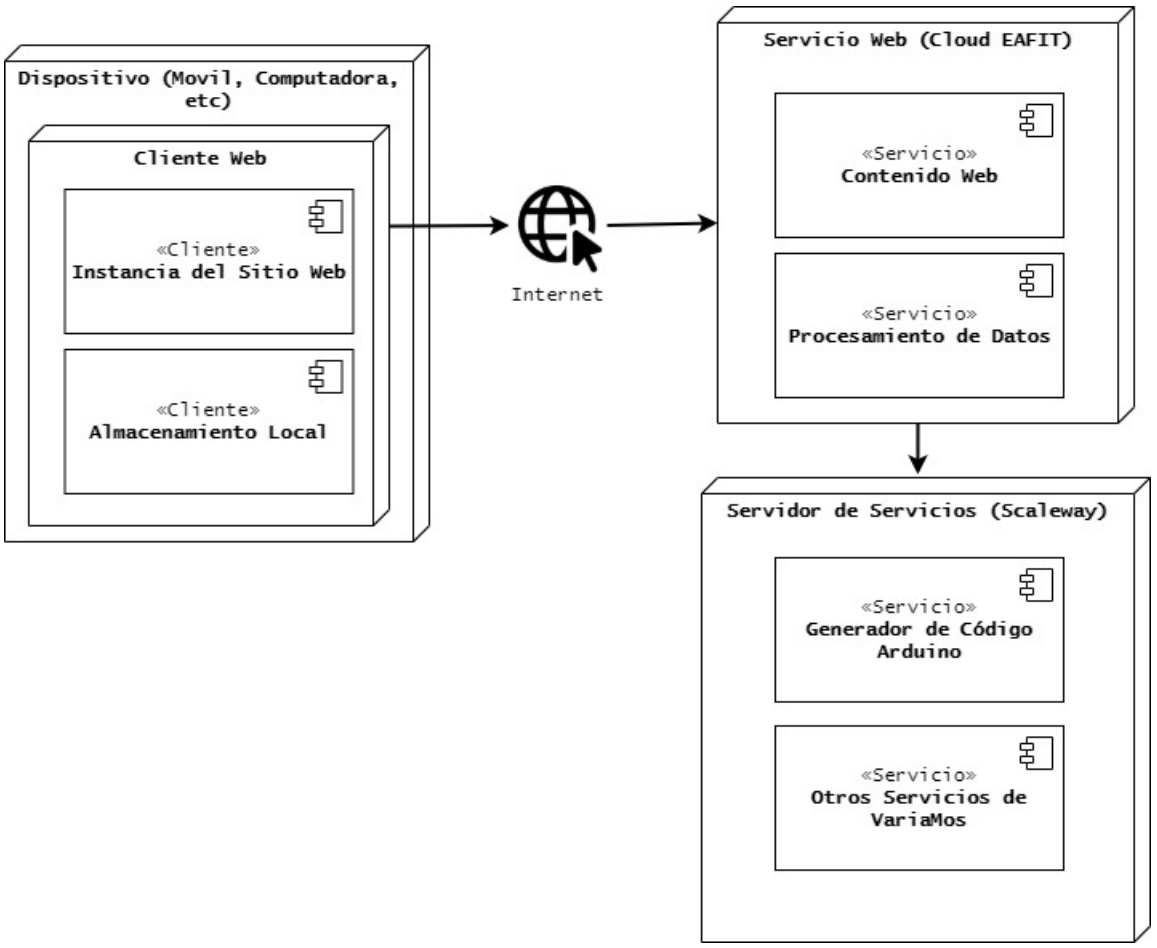


En el diagrama anterior se puede notar una división del proyecto en 4 módulos importantes, estos son:

- **Usuario:** Este modulo posee todos los componentes que hacen parte del cliente en general, siendo éstos parte fundamental de las vistas generales, ya que el aplicativo generalmente se encuentra 100% en el cliente y los datos relevantes son almacenados en el Almacenamiento Local, evitando así el uso de un backend que administre las acciones básicas del usuario.
- **Almacenamiento Local:** Este modulo hace referencia total al almacenamiento local de cada Usuario, basándose entonces en un almacenamiento meramente del cliente que utilice el usuario, ya que el aplicativo en su totalidad (exceptuando los servicios), son almacenados aquí.
- **Aplicativo Web:** Este modulo aunque sigue haciendo parte del cliente, implica toda la funcionalidad lógica que se procesa en base a las acciones del cliente durante las vistas. Se compone entonces de interacciones con el usuario y de procesamientos de datos y diagramas, encargándose también del envío de peticiones y comunicaciones con el servidor de servicios.
- **Servidor de Servicios:** El servidor de Servicios, también conocido como "*VariaMos Services*", es el backend encargado de dar funcionalidad los sistemas donde la carga en el cliente puede ser muy compleja y disminuir el rendimiento de la plataforma, por tanto, los servicios procesan esto en un servidor aparte y la información es devuelta al cliente dándole la oportunidad de obtener los resultados de su diagramado.

> Con base en estos modulos se realiza todo el proceso del componente de diagramado de proyectos Arduino incluyendo el proceso de generación de código y las etapas.

4.3. Vista de Despliegue del producto



En el diagrama de despliegue se dan los 3 elementos importantes que permiten la funcionalidad completa de la plataforma, en este caso se tiene el Cliente, el cual es el punto de partida y representa en sí al usuario, su dispositivo y su navegador, posteriormente se encuentra el Servicio Web el cual se despliega en un servidor (en este caso está siendo desplegado en los servidores de EAFIT), y finalmente el servidor de servicios, que está siendo desplegado por nosotros en una plataforma llamada Scaleway.

Links Últiles:

- Servicio Web: http://variamos.dis.eafit.edu.co/
- Servidor Backend: http://51.15.209.191:8080/
- (Versión antigua) Sub-Servidor Front-end: http://51.15.209.191:8081/variamosweb/
- Scaleway Cloud: https://www.scaleway.com/

> Los links que usan la IP numérica `51.15.209.191` hacen parte de nuestro servidor privado virtual que utiliza **Scaleway Cloud**, allí tenemos una versión 100% funcional de VariaMos basada en el fork de nuestros Github (estos no poseen la última versión del software puesto que este se actualiza todos los días y el ritmo de trabajo es más pesado en ese sentido, tenemos una copia alterna y estable).

5. Modelo de Datos

Como ya se ha comentado, el modelo de datos de variamos funciona con una persistencia única por archivos, los cuales se almacenan en el formato XML y también se tiene un almacenamiento local, que en los navegadores es denominado como `Local Storage`, es así pues como estos dos componentes son parte de la persistencia total del proyecto.

1) Componente: Local Storage

Local Storage	
PK	LocalStorage_ID
	Aplication Name
	Filetree
	domain_implementation_main_path
	domain_implementation_git_token
	domain_implementation_user_login
	cache_selectedApplication
	Filetree_User
	Filetree_ActiveTab

En este componente se tiene una breve descripción de cada campo:

- **LocalStorage_ID:** Esta ID es interna de cada navegador y se usa para identificar el almacenamiento local en la sesión concurrente del programa.
- **Aplication Name:** Aquí se almacena el proyecto y se usa el nombre del proyecto como el nombre del campo.
- **Filetree:** Este almacena la cantidad de pestañas del modelo que se muestra en ese momento.
- **domain_implementation_main_path:** Aquí se almacena el link endpoint del backend, el usuario configura este atributo desde el menú de la plataforma.
- **domain_implementation_git_token:** Aquí se almacena el token de github utilizado para el futuro despliegue del backend, el usuario configura este atributo desde el menú de la plataforma.
- **domain_implementation_guser_login:** Aquí se almacenan las credenciales del usuario de github utilizado para el futuro despliegue del backend, el usuario configura este atributo desde el menú de la plataforma.
- **Filetree_User:** En este se almacena toda la información referente a las acciones del usuario sobre los diagramas, información tal como: su selección actual, la pestaña que está viendo, etc.
- **Filetree_ActiveTab:** En este componente se refina la información de `Filetree_User` obteniendo únicamente la pestaña activa.

Ejemplo del almacenamiento local (Imagen tomada el 29/05/2020 con los últimos cambios oficiales):

Key	Value
Application - test - 1	<mxGraphModel> <root> <mxCell id="0"/> <mxCell id="feature" parent="0" visibl...
Filetree|model_component_index	8
domain_implementation_main_path	http://51.15.209.191:8080/
domain_implementation_git_token	[GITHUB TOKEN]
domain_implementation_user_login	[GIT USER LOGIN]
cache_selectedApplication - test - 1	[]
Filetree|User1	[{"children":[],"data":{"open":true,"isSelected":false,"level":1,"nodeId":1,"nodeName":"t...
Filetree|activetab	undefined

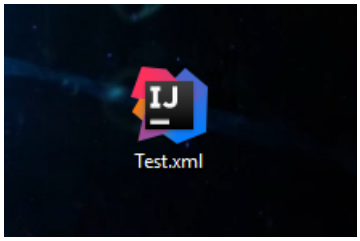
```
<mxGraphModel>
  <root>
    <mxCell id="0"/>
    <mxCell id="feature" parent="0" visible="0"/>
    <mxCell id="component" parent="0" visible="0"/>
    <mxCell id="binding_feature_component" parent="0" visible="0"/>
    <mxCell id="istar" parent="0" visible="0"/>
    <mxCell id="classdiag" parent="0" visible="0"/>
    <mxCell id="adap_architecture" parent="0" visible="0"/>
    <mxCell id="adaptation_hardware" parent="0" visible="0"/>
    <board type="board" label="ArduinoUNO" boardType="ArduinoUNO" subtype="ArduinoUNO" digitalPins="D2,D4,D7,D8,D12,D13" analogPins="A0,
      <mxCell style="shape=board" parent="adaptation_hardware" vertex="1">
        <mxGeometry x="230" y="180" width="280" height="180" as="geometry"/>
      </mxCell>
    </board>
    <D2 label="D2" type="digital" id="2">
      <mxCell style="shape=ellipse;perimeter=ellipsePerimeter;direction=north;movable=0;resizable=0;" parent="1" vertex="1">
        <mxGeometry x="10" y="24" width="7.5" height="7.5" as="geometry">
          <mxPoint as="offset"/>
```

2) Componente: Archivo XML

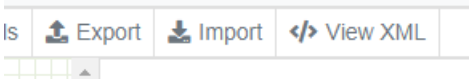
Este componente hace parte de la persistencia por archivos, donde el usuario puede descargar sus diagramas en el formato XML para posteriormente integrarlos importando el archivo ya sea en otro dispositivo o pasándoselo a sus compañeros de trabajo. Básicamente es una alternativa que inhibe a la plataforma de desarrollar una persistencia única para los usuarios ahorrando costes.

Prueba de Funcionamiento:

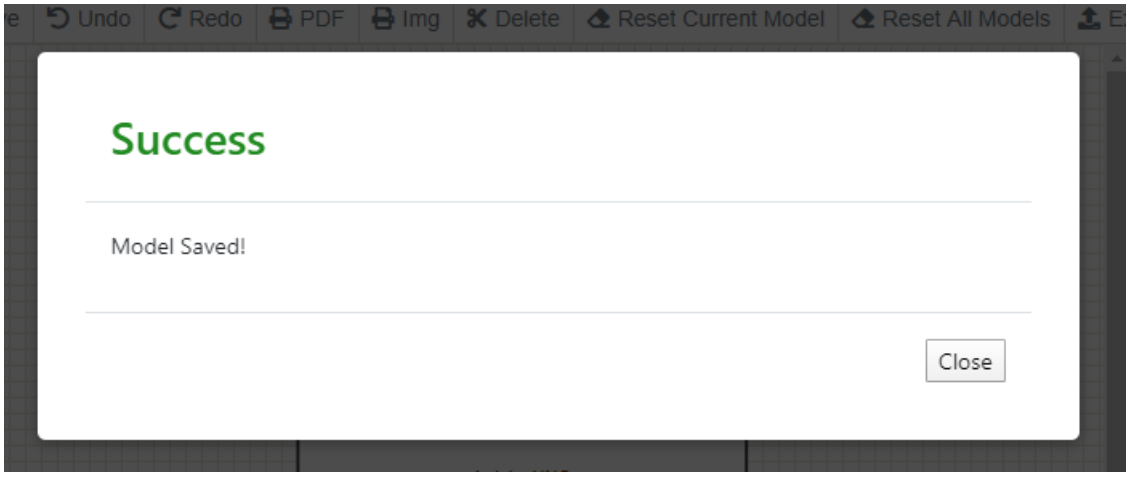
> Archivo Generado y Descargado.



> Barra de Herramientas.



> Mensaje que se muestra durante la carga y descarga de un proyecto.

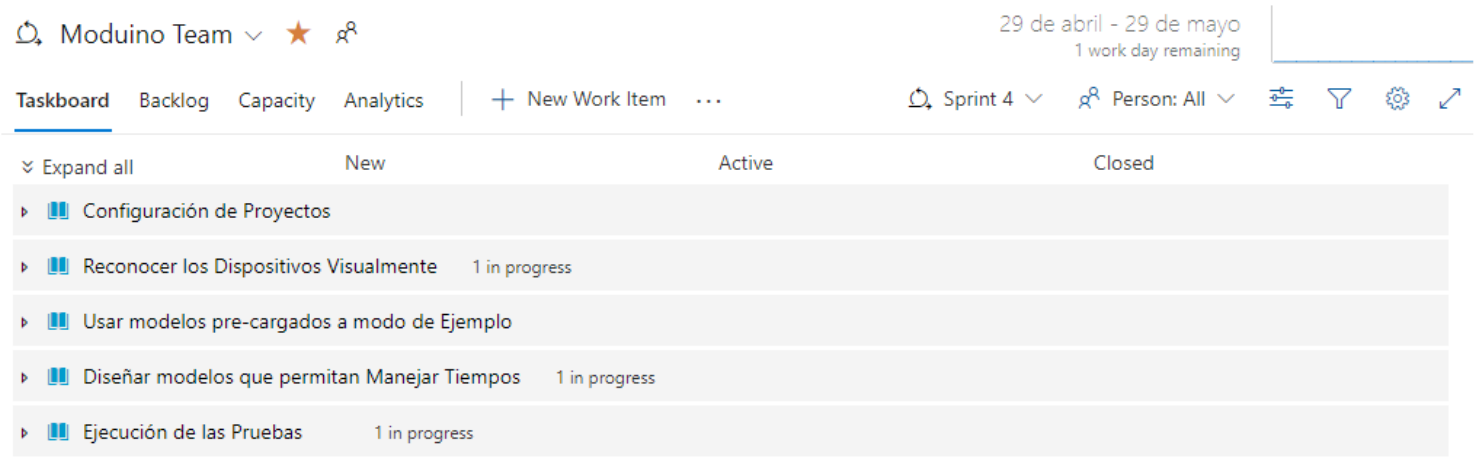


Es importante decir que el como se guardan estos modelos es basado en una librería de terceros denominada `mxGraph`, es gracias a esta librería y su modulo conversor en tiempo real que permite que todos los componentes gráficos tengan una versión textual capaz de ser utilizada de manera bidireccional obteniendo así nuevamente la versión gráfica.

Toda la información que implica está conversión se basa en el modulo `mxCodec` y toda la referencia y documentación a esta información se encuentra aquí: https://jgraph.github.io/mxgraph/docs/js-api/files/io/mxCodec-js.html

6. Historias de Usuario

En el siguiente link estaran ubicado las historias de usuario:
https://dev.azure.com/amlorav/Moduino/_sprints/taskboard/Moduino Team/Moduino/Sprint 4



7. Reuniones Semanales

Las reuniones semanales fueron alrededor de 3 (la gran mayoría fue con el grupo de trabajo y el P.O), sin embargo hay muchas sub-reuniones que se hicieron a lo largo de las semanas donde realizábamos pruebas de usabilidad a los participantes.

Estas reuniones se han hablado temas importantes tanto el progreso sustancial de las pruebas usabilidad, como también hemos tratado temas para la entrega final, implementacion de timer y cambios en el desarrollo de VariaMos Web en su imagen corporativa.

Esta comunicación ha sido constante por reuniones virtuales.

Las reuniones están en la pestaña Reuniones de la Wiki de Moduino y están compuestas por comentarios de cada uno de los integrantes del proyecto respondiendo las preguntas de las reuniones generales con metodología SCRUM. Que son:

¿Que hice?
¿Que dificultades tuve para hacer eso?
¿Que haré?

A continuación se adjunta el link directo de las reuniones
https://dev.azure.com/amlorav/Moduino/_wiki/wikis/Unknow-Project.wiki/7/Reuniones-Semanales

8. Repositorio

Actualmente Moduino se mantiene igual que antes, debido a que en cuanto a repositorios actualmente el proyecto se compone de 4 repositorios. Dichos repositorios son fundamentales y cada uno será explicado de manera breve a continuación:

- `moduino` (Link): Este repositorio tiene todo el contenido del generador de código que se lleva trabajando desde hace casi 10 meses, dicho generador está realizado en Java y su función es hacer uso de un metamodelo como componente principal para la interpretación dinámica de objetos C/C++/Arduino Lang, como piezas de código, convirtiendo entonces objetos dinámicos de Java en código de Arduino adaptable al sistema, el cambio general con la anterior versión es que se han solucionado alrededor de 10 bugs que generaban código con fallas y se introdujeron cambios importantes para poder agregar los nuevos componentes del desarrollo.

- `variamos-code-binder` (Link): Este repositorio permite interpretar la información recibida desde el Front-End (variamos-web), y haciendo uso del generador (moduino), producir el código y reenviarlo como respuesta. Se integra actualmente como un servicio al servidor de Spring-Boot (variamos-services). A su vez, este repositorio también fue cambiado de nombre, permitiendo un mejor entendimiento y se realizaron refactorizaciones en clases y presentación de los repositorios.

- `variamos-web` (Link): Este repositorio contiene todo el desarrollo de Variamos en cuanto al Front-End, donde se utiliza como framework principal VueJS y la libreria de graficos MxGraph, para poder producir un editor de diagramas con una paleta y demás funciones y características que ofrece la plataforma a los usuarios.

- `variamos-services` (Link): Este repositorio contiene todo el desarrollo de Variamos en cuanto al Back-End, donde se utiliza Java y el framework Spring-Boot para por medio de servicios integrados manejar una API que permita procesar datos y devolver respuestas que sean complejas o que requieran de un entorno Backend. Por parte de Moduino, utilizamos el Backend para poder integrar el (code-binder) y así utilizarlo desde el FrontEnd (variamos-web).

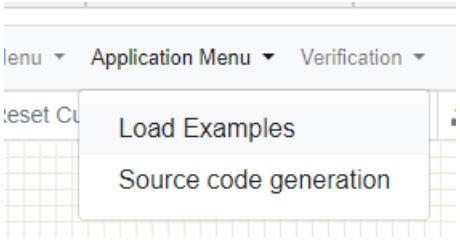
Todos estos repositorios se encuentran en la plataforma de Github, y cada uno es parte fundamental de todo el proyecto puesto que componen el trabajo incluyendo las metas y objetivos de Moduino. Además como estamos a su vez apoyando el trabajo que ya se venía realizando desde antes, son repositorios que ya tienen un desarrollo previo y nuestra intención es permitirnos a todos los integrantes tanto de Moduino como de VariaMos un fácil acceso a nuestras copias de los repositorios originales para poder ser integrados y desplegados en la plataforma oficial (http://variamos.dis.eafit.edu.co/#/).

9. Perspectivas

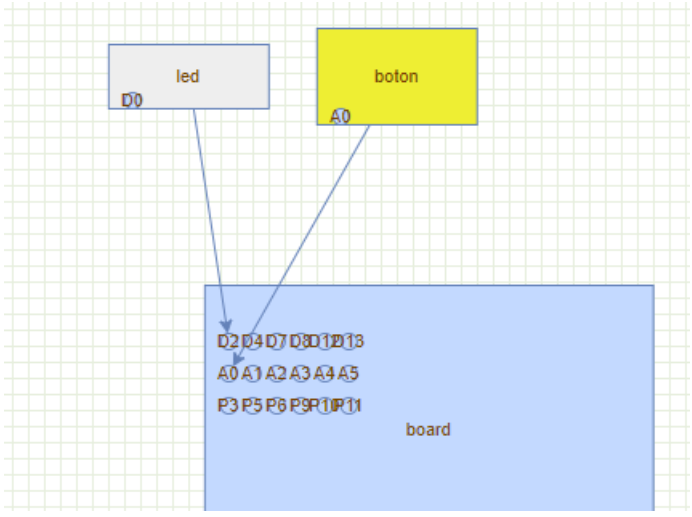
Para la solución propuesta, también se llevo a cabo un proceso de pruebas de usabilidad, donde buscamos determinar directamente desde la vista del usuario, la usabilidad de la aplicación web, más específicamente, la amigabilidad del frontend con el usuario, esto nos sirve para que ya sea Variamos, o nosotros mismos, sigamos trabajando en el proyecto. Esto nos dejó algunas ideas adicionales para sumarle a este proyecto, como lo es una modificación del frontend, donde el diagramado sea más cercano al usuario y las imágenes de los componentes sean las que vemos en la vida real, para quitarle un poco la abstracción al proyecto y que el estudiante sienta la plataforma mucho más cercana y fácil de usar. También planteamos como una propuesta a futuro que este generador de código basado en modelos, no sólo sea útil para arduino, sino que lo puede ser para muchos más lenguajes de programación como por ejemplo Java, C++ y Python. Si esto se lograra, estaríamos dando un paso muy grande en el mundo de la programación, ya que se aceleraría la creación de soluciones de software donde casi cualquier persona podría crear una solución real únicamente diseñando y modelando su propuesta de solución.

En función de las pruebas realizadas se han introducido cambios no oficiales pero si experimentales en el Frontend, estos cambios hacen parte de ramas aún no aplicadas y se espera que sean aprobados pronto.

1. Adicional de la opción de cargar ejemplos.



2. Colores por dispositivo (mejor identificación).



10. Referencias

[1] New Bare Metal Offer (n.d.). [online] Recuperado de: https://www.scaleway.com/en/ .

[2] Anon, (2020). [online] Recuperado de: https://www.quora.com/Which-are-the-alternatives-to-UML .

[3] VariaMos – Families of Systems & SAS Modeling Tool. (2020). Recuperado de: https://variamos.com/home/ .

[4] DBMS Database Models. (2020). Recuperado de: https://www.studytonight.com/dbms/database-model.php .

[5] Software Architecture Guide. [online] Recuperado de: https://martinfowler.com/architecture/ .

[6] Github Repositories. [online] Recuperado de: https://developer.github.com/v3/repos/ .

[7] Story Mapping, Visual Way of Building Product Backlog. (2015). Recuperado de: https://www.thoughtworks.com/insights/blog/story-mapping-visual-way-building-product-backlog

[8] IBM Knowledge Center. (2014). Recuperado de: https://www.ibm.com/support/knowledgecenter/SS8PJ7_9.5.0/com.ibm.xtools.modeler.doc/topics/cdepd.html

[9] UML Deployment Diagram. (2016). Recuperado de: https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-deployment-diagram/

Entregable en video

https://youtu.be/Fo21Fl8GTGQ

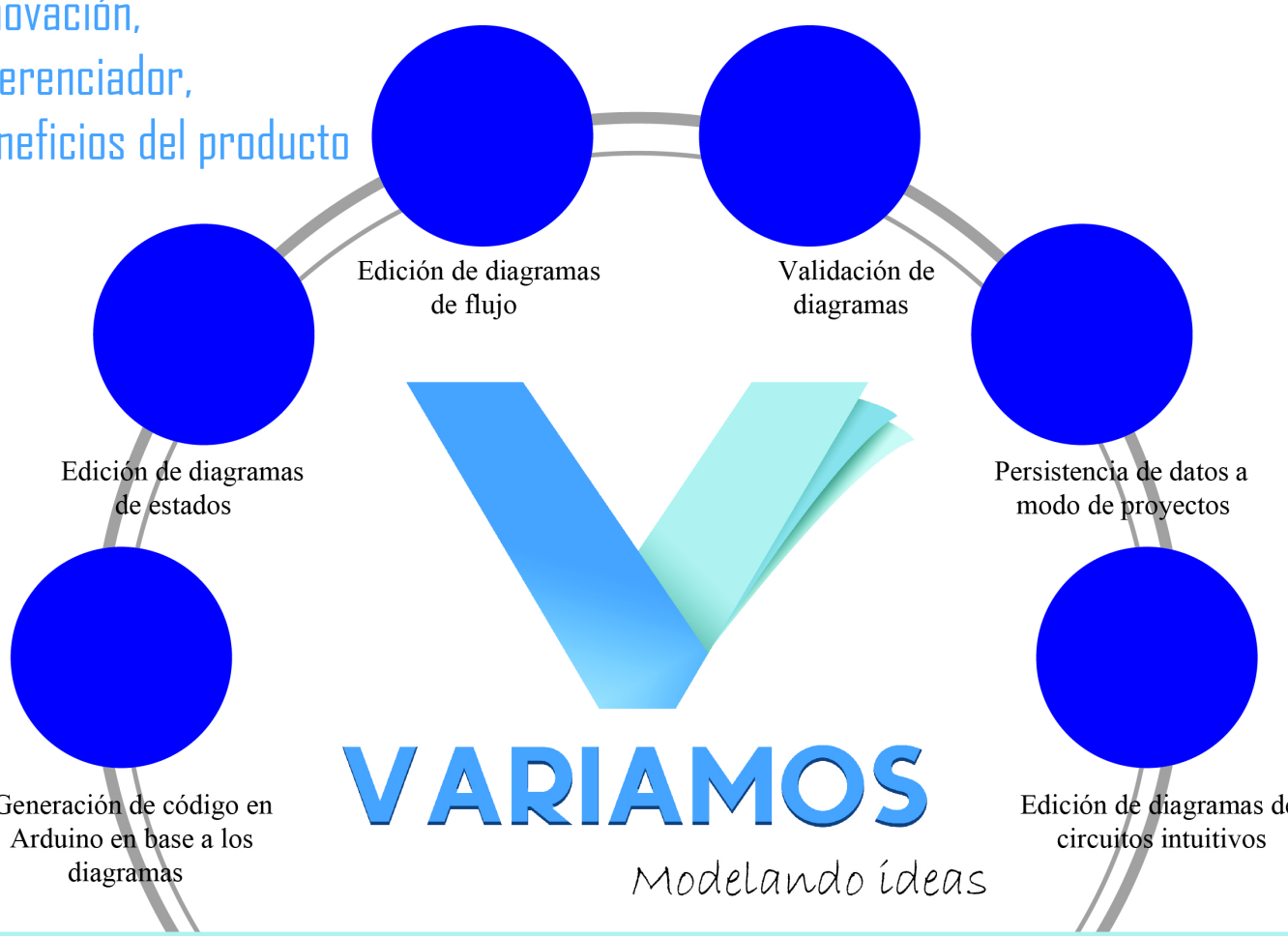
Poster

MODUINO

PROBLEMA QUE RESUELVE

Durante el desarrollo de proyectos en Arduino, es muy común observar que los estudiantes suelen presentar inconvenientes a la hora de codificar, y son reincidentes en tener errores y retrasos implementando así la repetitividad de código o buscando soluciones a base de prueba y error.

Innovación, Diferenciador, Beneficios del producto



PRODUCTO

Moduino es un proyecto que colabora con la plataforma web VariaMos donde su enfoque es proveer herramientas dedicadas al desarrollo de diagramas de circuitos, estados y de comportamiento por medio de flujos, para proyectos arduinos, en este el desarrollador no necesitará codificar si no diagramar modelos para al final obtener un código totalmente funcional en arduino.

Autores + Afilición: Abraham M. Lora - Daniel G. García - Mateo Montes L.- VariaMos

Resultados de usabilidad

usabilidad.pdf

Test de usabilidad de Variamos web enfocado en el modelado de proyectos Arduino

ISO/IEC 25062 Common Industry Format for usability Test Reports

Variamos web enfocado en el modelado de proyectos Arduino

Test por Abraham Lora V. – Daniel García G. – Mateo Montes L.

Marzo 2020

Email: amlorav@eafit.edu.co - mmontesl1@eafit.edu.co - dgarciag@eafit.edu.co

Medellín - Colombia

Resumen.

El equipo Moduino realiza una investigación en la cual trabajara con una parte del sistema VariaMos el cual se basa en el modelado de proyectos de Arduino y por medio de este lograr la generación de código automático, también se llevara a cabo una investigación con estudiantes de pregrado de la universidad EAFIT para saber qué tan intuitivo, usable y amigable es esta.
Se diseñó una prueba de usabilidad siguiendo el formato común de la industria ISO/IEC 25062:2006 para las pruebas de usabilidad en la cual ocho estudiantes de Pregrado de la universidad EAFIT del programa ingeniería de sistemas y carreras con relación a esta, participaran en unas pruebas, dándole unas tareas a realizar mediante una serie de pasos en la plataforma.

1. Introducción

En el presente se realizará un experimento mediante de la implementación de un experimento el cual mediante un paso a paso que deben seguir de un modelado para proyectos Arduino para luego hacer la recolección de datos.

Por medio de imágenes se buscará el efectivo uso del participante a la hora de la interacción con la plataforma de VariaMos Web para su objetivo final, el cual es lograr la correcta generación de código automático Arduino.

El usuario objetivo serán los estudiantes de primeros semestres que ven materias relacionadas con programación Arduino como lo son electrónica digital y conmutación de computadores; además este experimento será con el fin de concluir que tan amigable, usable y efectiva es la plataforma en el sentido de modelado vs codificación tradicional.

En el numeral 2 se presenta una prueba de usabilidad de VariaMos web basada en el modelado de proyectos Arduino.

2. Pruebas de usabilidad

En esta sección se presentará las pruebas de usabilidad de Variamos web enfocado en el modelado de proyectos Arduino y la idea principal es probarla en estudiantes de pregrado de los programas de ingeniería de sistemas y carreras con relación a esta, para así comprender lo fácil o difícil que es seguir y entender el enfoque que se ha estado llevando en cuanto a las líneas de producto de VariaMos. Para guiar la prueba de usabilidad, definimos la siguiente pregunta de investigación.

RQ: ¿Variamos web es una herramienta intuitiva, efectiva y fácil de manejar?

En esta evaluación se decidió desarrollar y realizar una prueba de usabilidad utilizando el Formato Industrial Común ISO/IEC (CIF) para pruebas de usabilidad. También se aplicó tres técnicas de evaluación: (I) una para la definición de las tareas experimentales, (II) otra para evaluar la satisfacción del usuario mediante la recopilación de su opinión a través de una encuesta, y finalmente (III) una entrevista semiestructurada para enriquecer esta prueba de usabilidad.

2.1 Procedimiento

La prueba de usabilidad fue diseñada como un proceso con siete actividades (véase la Fig. 7), que se describe a continuación.

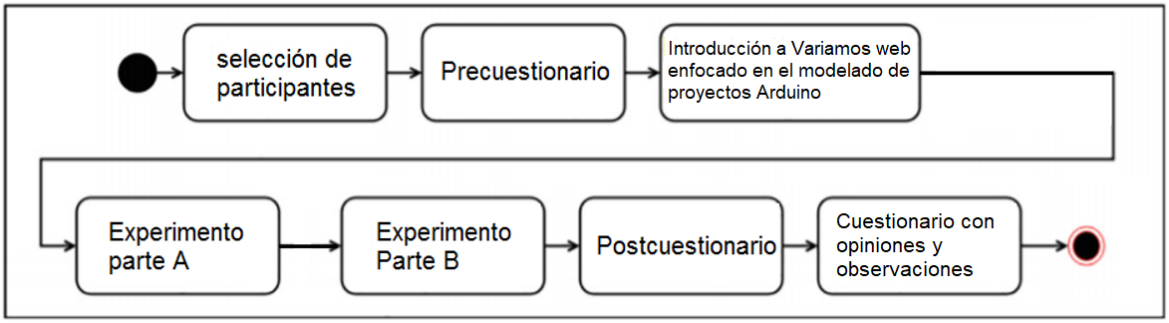


Fig 7. Proceso de prueba de usabilidad (diagrama de actividad UML)

Selección de participantes.

Se realizará una selección de mínimo ocho estudiantes de Pregrado de la universidad EAFIT del programa ingeniería de sistemas y carreras con relación a esta.

La prueba de usabilidad fue diseñada para una sesión de 50 minutos y como máximo 1 hora. Estos participantes se clasifican como “Estudiantes de ingeniería de sistemas y carreras con relación a esta que están dispuestos en probar la usabilidad de Variamos web el cual está enfocado en el modelado de proyectos Arduino".

Material entregado.

Se le enviara a la persona que accedió a realizar el experimento un archivo .rar con el nombre de experimento.rar el cual se debe descomprimir dándole clic derecho “extraer aquí” (recuerde que el archivo debe estar en el escritorio para fácil acceso); Adicional a esto se realizara una videollamada a la hora de hacer el experimento y por este medio el administrador de este transmitirá un video de Introducción a Variamos web enfocado en el modelado de proyectos Arduino.

Cuestionario previo (5 min).

Se solicita a los participantes que complementen un cuestionario previo relacionado con sus antecedentes y experiencia en Arduino. El cuestionario anterior se elaboró utilizando una escala Likert, que tenía un formato de cinco puntos: 1) muy en desacuerdo, 2) algo en desacuerdo, 3) ni está de acuerdo ni en desacuerdo, 4) algo de acuerdo, y 5) muy de acuerdo.

Introducción a Variamos web enfocado en el modelado de proyectos Arduino (10 min).

Se diseño un archivo por medio del cual se muestran imágenes de las herramientas que se tiene en la plataforma de VariaMos Web, esta introducción será importante porque los participantes que no tengan conocimiento en la plataforma de VariaMos Web, tendrán una base e introducción de la plataforma para el modelado de los ejemplos.

Experimento Parte A (Límite: 15 min).

Se le pedirá al estudiante que realizara el experimento que descargue la carpeta enviada por correo institucional el cual contiene unos documentos para la realización de la prueba. esta carpeta será ubicada en el escritorio con el nombre EjemploVariamos, acá se encontrarán 2 subcarpetas la cual se ingresará dando doble clic en la que dice “test1”, en esta carpeta se encontrara 2 archivos, uno el cual es un documento con el nombre “test1variamos.pdf” y otro que se llama “linkvariamos.url”

En el primer archivo(test1variamos.pdf), se encontrará un tutorial por medio de imágenes el cual busca el encendido y apagado de un led.

El segundo archivo(linkvariamos.url) te dirigirá a la plataforma VariaMos web al darle doble clic. El estudiante realizará el experimento con el paso a paso del documento, adicional a eso se tendrá un (1) administrador para la prueba que estarán observando y atendiendo las dudas que surjan durante el experimento.

Experimento Parte B (Límite: 15 min).

Los estudiantes comenzaran a completar la tarea 2 la cual está ubicada en la segunda subcarpeta llamada “test2” donde se ingresará dando doble clic en la que dice, en esta carpeta se encontraran 2 archivos, uno el cual es un documento con el nombre “test2variamos.pdf” y otro que se llama “linkvariamos.url”.
En el primer archivo (test2variamos.pdf), se encontrará un reto el cual teniendo en base lo realizado en el experimento “Parte A” sé realizara la conexión de un segundo LED para representar el encendido de un LED y el apagado del otro LED esto debe suceder a la vez. El segundo archivo (linkvariamos.url), te dirigirá a la plataforma VariaMos web al darle doble clic.

Cuestionario posterior (5 min).

Los participantes serán sometidos a un cuestionario posterior, el cual incluirá preguntas sobre: i) factores externos, ii) la satisfacción general, iii) el rendimiento de Variamos Web enfocado en el modelado de Proyectos Arduino y un apartado para dejar comentarios.

Cuestionario con opiniones y observaciones

Se preguntará al participante cuatro preguntas abiertas sobre la usabilidad de la herramienta, para luego guardar las respuestas de los participantes. Las preguntas son: (i) ¿Qué te gustó de VariaMos web basada en el modelado de proyectos Arduino? (ii) ¿Qué te disgustó/Qué debería mejorarse? (iii) ¿Qué se te dificulto a la hora de realizar las tareas en la plataforma de VariaMos web basada en el modelado de proyectos Arduino? y (iv) ¿Qué se podría mejorar visualmente para que el desarrollo de los diagramas sea entendible?