



UNIVERSIDAD
DE LOS ANDES

Propuesta de Proyecto de Grado

Presentado ante la ilustre Universidad de Los Andes como requisito parcial para
obtener el Título de Ingeniero de Sistemas

GeMVP. Diseño y Construcción de un Gestor de Producto
Mínimo Viable para una Startup.

Por

Br. Gustavo E Briceño F.

Tutor: Dr. Gerard Páez.

Febrero 2017.

©2017 Universidad de Los Andes, Mérida, Venezuela

Capítulo 1

Introducción

1.1 Planteamiento del Problema

La universidad venezolana se abre a tiempos de cambio. Actualmente, uno de los aspectos más relevantes en el campo educativo es la creatividad en periodos cruciales para consolidar proyectos que tengan un gran significado para todos. Por ello, la Universidad de Los Andes consagrada en cumplir su papel histórico de promover la transformación en la gestión del conocimiento y del pensamiento, cada día ha estado más comprometida en su rol para lograr la verdadera dimensión y pertinencia social. Los estudiantes, profesores, empleados y autoridades son todos protagonistas del aprendizaje en una organización más humanizada y avanzada, representando una reserva de talento, capacidad pedagógica, moral y valores éticos. De esta manera, se aspira a la nueva universidad por vía de ideas bien concebidas y administradas con inteligencia dentro de un marco de pluralidad y diversidad; generando conocimientos e incorporando y creando nuevas tecnologías destinadas a una sociedad más globalizada y competitiva.

A día de hoy y conscientes de las necesidades de la sociedad actual y como una forma de dar soluciones efectivas, se impone un proceso de transformación de la universidad, en una época donde la innovación y el emprendimiento se unen para dar respuesta a los problemas. En esta nueva perspectiva, se gestionan programas y proyectos que permiten revitalizarla como organización; asimismo, los estudiantes son capaces de desarrollar tecnologías mientras aplican los conocimientos obtenidos en su carrera universitaria, construyendo así un producto escalable y de interés común para muchos. En este proceso de cambio, surge la necesidad de orientación y asesoría en todos los aspectos que conllevan a la ejecución de una idea. Otro aspecto clave está representado por todas las herramientas que permitan organizar, visualizar cada etapa y acelerar el desarrollo del proyecto. Lo planteado tiene implicaciones importantes, con especial atención en la construcción de una plataforma tecnológica para el emprendimiento universitario con software especializado en la asesoría legal, valoración de métricas, modelo de negocios, conexión con otros inversionistas o innovadores y gestión de tareas.

Esta combinación de herramientas proporciona apoyo a la creación de un ambiente de innovación y desarrollo, dando soporte al programa ULA startups; este último constituyéndose como el pilar fundamental de la gestión y aceleración de las ideas cuyo génesis sea desde la universidad. A continuación se muestra los proyectos de grados que conforman la plataforma en general, han sido integrados como parte de una sola estructura:

- Diseñar, Desarrollar e Implementar un Sistema de Información Web para la Gestión de Datos de ULA Startups (Gil, A; 2017).
- Aplicación en Javascript para el Desarrollo Startups (Cuevas, R; 2017).
- Aplicación Web para la Gestión de Documentos Legales de las Startups (Espinoza, L; 2017).
- GeMVP. Diseño y Construcción de un Gestor de Producto Mínimo Viable para una Startup (Briceño, G; 2017).
- Desarrollo e Implementación de un Algoritmo para Valorar y Jerarquizar un Conjunto de Emprendimientos Tecnológicos de Arranques (Ramírez, F; 2017)

ULA startups podría ser el resultado de la ruptura de los paradigmas tradicionales de la universidad, así como la integración de todos los sectores de la comunidad en los procesos de cambio y concepción de la nueva universidad.

En un mundo donde las tecnologías crecen con una velocidad atroz, los emprendimientos están a la orden del día. Según el Global Entrepreneurship Research Association (GERA), el emprendimiento ha crecido a nivel mundial, estamos hablando de que en algunos países latinoamericanos, una de cada seis personas tiene un emprendimiento o monta su propia empresa. Diversos estudios en muchos países han concluido lo mismo, el fracaso de los emprendimientos a nivel global es alarmante, se habla de un 80% de fracaso a nivel global como menciona el doctor Claudio Soriano en su artículo “Por qué fracasan las Pymes” (2005) y en países como Venezuela hasta un 90% de fracaso para todas aquellas ideas que surgieron y se convirtieron en empresas.

Es importante acotar que existen emprendimientos de todo tipo, nosotros abarcaremos los emprendimientos de tipo tecnológicos. Una Startup es un emprendimiento tecnológico que surge mayormente con una idea novedosa, busca generar y comercializar un producto desarrollado apoyándose altamente en la tecnología y dejando de lado aquellos métodos tradicionales de crear una empresa siguiendo un plan de negocios rígido. Para desarrollar dicho producto se empieza con una primera versión que cuenta con menos características del producto final, esta primera versión es lo que se llama Producto Mínimo Viable (MVP por sus siglas en inglés), y es una parte fundamental del Startup ya que con el MVP pueden evaluar muchos factores que ayudarán de manera positiva la creación de su producto final.

En la actualidad dentro de la cultura Startup, se vive un mundo de incertidumbre y alto dinamismo, donde según Peris R, (2014) más del 75% de las Startups fracasan, y lo hacen por diversos motivos, uno de ellos, como dice Gutiérrez I, (2013) es el no llevar a cabo la idea base del Startup, y terminando como una idea ejecutada por otros.

Para este problema, se propone desarrollar un Gestor de Producto Mínimo Viable (GeMVP) con el objetivo de contribuir de manera positiva con la ejecución de la idea de cada Startup y así sumar a sus probabilidades de éxito.

1.2 Justificación

El diseño y la construcción del GeMVP es sumamente importante para apoyar uno de los momentos más importantes de la vida de una Startup como es la construcción del Producto Mínimo Viable (MVP), ya que según Gutiérrez I, (2013) una idea por sí sola sin ejecutarla es equivalente a una Startup fracasada y es conocido en el mundo que más del 75% de las Startups fracasan. Las Startups viven en continua incertidumbre y son altamente dinámicas, GeMVP proporcionará una herramienta que se adapta a esta realidad, y ayude a llevar a cabo cualquier MVP sin importar su naturaleza.

Los temas de estudio que se tuvieron que abordar para llevar a cabo en este trabajo fueron, la cultura Startup, Emprendimiento, Planificación, Gestión de Proyectos, Programación Web, Diseño Web, Base de Datos, Estudio de Mercado, etc. Para el cumplimiento de los objetivos planteados se hizo el uso de la imaginación y la creatividad.

1.3 Objetivo General

Diseñar y Desarrollar una herramienta web que funcione como gestor de producto mínimo viable para las startups y contribuya de manera positiva con la ejecución de las ideas de las startups.

1.4 Objetivos Específicos

- Diseñar una herramienta web que busque funcionar como gestor de producto mínimo viable para las startups.
- Desarrollar la herramienta web que busca funcionar como gestor de producto mínimo viable para las startups.
- Evaluar el desempeño de la herramienta web.
- Hacer pruebas que permitan verificar si la herramienta contribuye de manera positiva con la ejecución de las ideas de las startups.

1.5 Alcance

Al revisar la literatura no se consiguió antecedentes del mismo tipo de problema que se pretende abordar en la presente investigación, por lo que la investigación y desarrollo iniciarán con carácter exploratorio, y servirá como base a futuros desarrollos que surjan a partir de esta herramienta.

1.6 Antecedentes

Peris R, (2014) en su trabajo de grado titulado “START-UPS TECNOLÓGICAS: EL RETO DEL CRECIMIENTO GLOBAL” define a los startups, así como todos los conceptos relacionados y muestra la importancia que ha adquirido la cultura Startup en los mercados punteros del mundo entero como un nuevo modelo de crecimiento social y económico, determinando cuales son los factores que influyen en el probable éxito o fracaso de las startups.

Por su parte Gutiérrez I, (2013) en su trabajo final de carrera titulado “Creación de un emprendimiento de software utilizando metodología Lean Startup” afirma que el valor de la idea de producto sin su desarrollo es un fracaso seguro, ya que lo que importa no es la idea en sí, sino su ejecución, así como también desarrolla un producto tecnológico utilizando una metodología llamada Lean Startup, la cual define la importancia de tener un producto mínimo viable (MVP), en busca de obtener un feedback temprano por parte del cliente antes del lanzamiento del producto principal.

Ahora Muñoz J, (2012) en su trabajo para optar al título de Ingeniero Civil Industrial en la Universidad de Chile titulado “Diseño de un programa de gestión para la evaluación y rediseño continuo de un Startup” diseñó un programa de gestión para las startups que siempre se encuentran en constante cambio y evolución, utilizando diferentes metodologías y herramientas para abarcar la mayor cantidad de áreas posibles de la Startup como la alineación del equipo, declaración y compromiso de actividades y la medición de las tareas y el trabajo realizado; finalmente el modelo de gestión obtenido mezcla procesos y herramientas de management con procesos de aprendizaje personal, grupal y de procesos productivos.

Finalmente Fino D, (2013) en su ensayo para optar al título de Especialista en Alta Gerencia en la Universidad Militar Nueva Granada titulado “Innovación en modelos de negocio: Metodología Lean Canvas en una Startup de base tecnológica” busca precisar el origen, los avances, las ausencias, las posibilidades y las perspectivas de los tantos y diferentes modelos de negocio que actualmente existen y son utilizados por las empresas del siglo XXI especialmente las startups, para luego innovar sobre un modelo de negocios innovador y obtener el Lean Canvas, mostrando la gran diversidad de modelos de negocios existentes y haciendo referencia implícita a la gran

incertidumbre que viven los startups y que cada uno de ellos puede afrontar sus problemas de una forma diferente.

1.7 Metodología

La metodología utilizada para el desarrollo del GeMVP será el modelo en espiral propuesto por Barry Boehm en 1986, las razones parecen obvias, el modelo en espiral proporciona el material para el desarrollo rápido de versiones incrementales de software y se puede describir de la siguiente manera:

El modelo de desarrollo en espiral es un generador del modelo de proceso guiado por el riesgo que se emplea para conducir sistemas intensivos de ingeniería de software concurrente y con múltiples usuarios. Cuenta con un enfoque cíclico para el crecimiento incremental del grado de definición e implementación de un sistema, mientras disminuye su grado de riesgo.

En fin, esta metodología nos permite ir refinando el modelo de software en cada iteración y que las versiones se acerquen a la versión final y estable de manera incremental, evaluando las alternativas para llegar a este punto y eligiendo aquellas en la que el riesgo sea bajo y no comprometa la culminación del proyecto.

1.8 Cronograma de Actividades

Semana	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Reuniones con el tutor	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Entrega de la propuesta		X														
Presentación de avance										X						
Entrega del Proyecto de Grado															X	
Presentación del Proyecto de Grado							X									X

Tabla 1.1: Cronograma de Actividades

1.9 Cronograma de Evaluaciones

- Actividad 1: Revisión bibliográfica de tesis anteriores vinculadas con el tema de la investigación.

- Actividad 2: Revisión de tutoriales y bibliografías acerca del diseño y programación web.
- Actividad 3: Revisión de bibliografías acerca de planificación.
- Actividad 4: Diseñar la herramienta GeMVP.
- Actividad 5: Desarrollar la herramienta GeMVP.
- Actividad 6: Hacer las pruebas necesarias.
- Actividad 7: Documentar la información.

Actividad / Semana	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Actividad 1	X	X	X	X												
Actividad 2	X	X	X	X												
Actividad 3	X	X	X	X												
Actividad 4				X	X	X										
Actividad 5							X	X	X	X	X					
Actividad 6												X	X	X		
Actividad 7	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

www.bdigital.ula.ve

Capítulo 2

Marco Teórico.

2.1 *¿Qué es una Startup?*

Una Startup es una organización humana con gran capacidad de cambio, que desarrolla productos o servicios, de gran innovación, altamente deseados o requeridos por el mercado, donde su diseño y comercialización están orientados completamente al cliente. [1]

Desde nuestro punto de vista, podríamos definir una Startup como un emprendimiento tecnológico, el cual surge de una idea novedosa, que busca producir y comercializar un producto o servicio. Mayormente empieza con un producto mínimo viable (MVP), que no es más que el primer producto o servicio que se lanza al mercado, con la menor cantidad de características posibles y con la finalidad de conocer al detalle a los posibles clientes potenciales.

2.2 *¿Cómo está formada una Startup?*

Una Startup puede estar formada de diversas maneras, pero siempre se recomienda que empiece con al menos dos fundadores, o lo que llamaremos cofundadores, adicionales a estos, tenemos a los inversionistas que se dividen en clases de acuerdo al momento de su inversión y finalmente tenemos a los ingenieros o equipo de trabajo. Existen casos donde una Startup está formada por una sola persona, lo cual indica que no existe una regla rígida de cómo debe estar formada la Startup.

2.3 *¿Qué busca una Startup?*

La idea principal de una Startup es monetizar un producto o servicio, que busca suplir una necesidad, apoyada fuertemente en la tecnología. Comúnmente los productos o servicios ofrecidos buscan ser novedosos y hacerse tendencia tanto local como global.

2.4 ¿Qué es un MVP?

Un Producto Mínimo Viable o MVP por sus siglas en inglés, es la versión más pequeña (que carece de ciertas funcionalidades) del producto que busca comercializar la Startup, y el presente trabajo gira en torno al mismo.

2.5 ¿Cuán exitosas son las Startups?

La gran realidad a la que se enfrentan las Startups día a día no es muy prometedora, ya que según las estadísticas, más del 80% de las Startups fracasan, debido a múltiples razones.

Entre las principales razones tenemos que las Startups no conciben un MVP a tiempo para poder sacar al mercado y analizar su comportamiento.

2.6 ¿Qué es un Gestor de Proyectos?

Un gestor de proyectos es una herramienta o software que permite llevar un control minucioso de cada uno de los proyectos de una empresa. El control implica todas las partes del trabajo, como la planificación, desarrollo y producción, así como el trato con el cliente. [2]

GeMVP puede visualizarse como un gestor de grandes proyectos como pueden llegar a serlo cada Startup.

2.7 ¿Qué es un grafo dirigido?

Un grafo dirigido o dígrafo consiste de un conjunto no vacío de elementos V , llamados vértices o nodos y un multiconjunto de A de pares ordenados de V , llamados arcos o aristas. Si un dígrafo no tiene lazos ni arcos múltiples diremos que es un dígrafo simple. Si r y w son vértices de V , entonces un arco rw se dice que está orientado o dirigido de r a w . [3]

2.8 ¿Qué es una línea de tiempo?

Una línea de tiempo es una herramienta que se utiliza para registrar y ordenar datos cronológicos como fechas y períodos de tiempo de forma clara y sencilla. En ella podemos relacionar acontecimientos importantes y los personajes que participaron en ellos con la fecha en que ocurrieron. [4]

2.9 ¿Qué es una herramienta web?

Es una aplicación software que se codifica en un lenguaje soportado por los navegadores web en la que se confía la ejecución al navegador. [5]

2.10 ¿Qué es una aceleradora de Startups?

Es una institución para impulsar startups mediante un programa basado en convocatorias con un plazo de tiempo estipulado. Estos programas incluyen mentorización, formación intensiva, educación digital y tutorización por parte de la empresa. Todo el proceso concluye con un Demo Day o con un pitch público. [9]

Capítulo 3

Diseño, Requerimientos y Modelo.

3.1 Diseño del Sistema.

Para ayudar a las Startups con el desarrollo de su producto mínimo viable, se decidió diseñar una aplicación web con características específicas, como:

- Que sea fácil de usar
- Con buen tiempo de respuesta
- Cómoda
- Intuitiva
- Sencilla

Teniendo todo esto en cuenta, nos basamos en la teoría de grafos para representar todas las tareas que una Startup debe cumplir para llegar a un objetivo final, por lo que apareció nuestra primera área de trabajo llamada “WorkBoard”, que no es más que una pizarra digital donde cada Startup podrá dibujar mediante un grafo dirigido todas las tareas que debe ir cumpliendo para desarrollar su MVP.

Lo importante de esta pizarra, es que le permita dibujar a las startups de manera dinámica y sin salirse de una misma pantalla todas sus asignaciones de actividades por cumplir.

Seguidamente, para poder evaluar ya sea por el gerente de la Startup, un cofundador o un ente externo encargado de evaluar asignaciones, se creó un área de trabajo llamada “TimeLine”, que no es más que una línea de tiempo donde se pintan todas las actividades que tiene la Startup para desarrollar su MVP, con sus respectivas fechas de cumplimiento y es el lugar en donde se revisa el desarrollo de cada MVP para tomar decisiones relacionadas con tiempos de entrega, número de actividades, y cualquier otra sugerencia que pueda hacer el encargado de revisar el desarrollo del MVP.

Finalmente, para asignar cada tarea, debemos recolectar cierta información de la cual se detallará más adelante, se creó una ventana emergente dentro de la misma pantalla de trabajo, que recoge los datos correspondientes a cada tarea, como su título, su fecha de inicio, su fecha de culminación en caso de que tenga, entre otros.

A continuación se muestran las diversas figuras para representar el diseño en principio de la herramienta GeMVP:

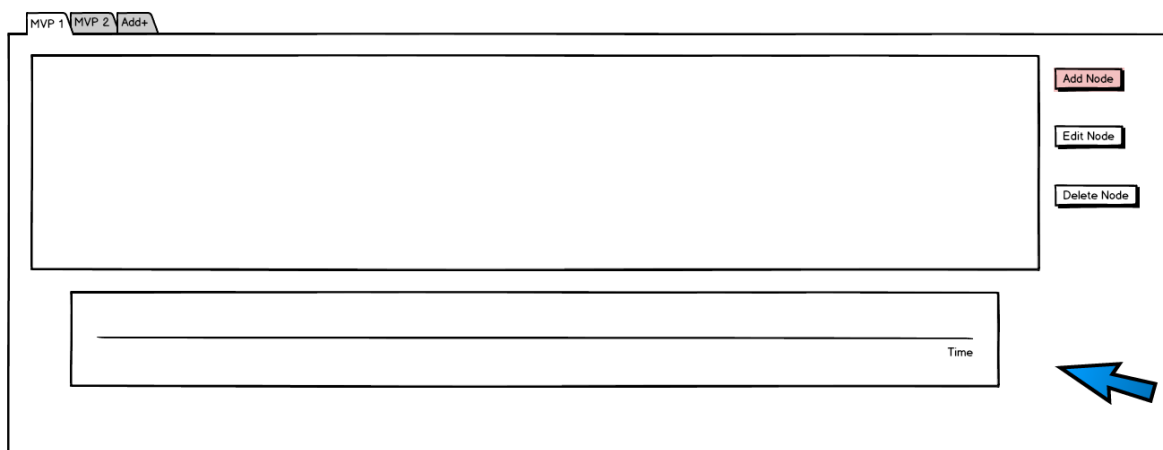


Figura 1. GeMVP

Este fue el diseño original de cómo se quería la página principal de la herramienta.

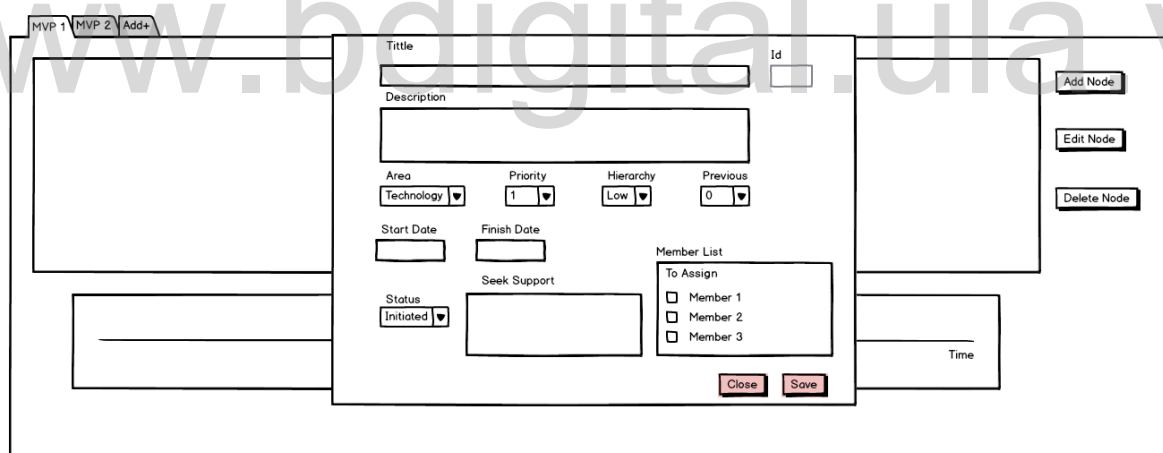


Figura 2. Añadir Nodo

Dentro de la página principal, encontramos una ventana emergente llamada modal, la cual permitiría ingresar los datos de cada hito, dentro de ella observamos los campos para ingresar título, descripción, área, prioridad, jerarquía, previo (quien es el hito predecesor), fechas de inicio y finalización, estatus, un recuadro para solicitar ayuda y los miembros asignados a desarrollar la actividad.

Figura 3. Editar Nodo

Para editar cada nodo, se debería abrir la ventana emergente con los datos pre cargados y con la funcionalidad de ser editables para posteriormente guardar la edición presionando el botón de guardar.

Figura 4. Eliminar Nodo

Una vez se seleccione el nodo que se quiere eliminar, se abre nuevamente y de manera automática la ventana emergente ahora con la opción de eliminar.

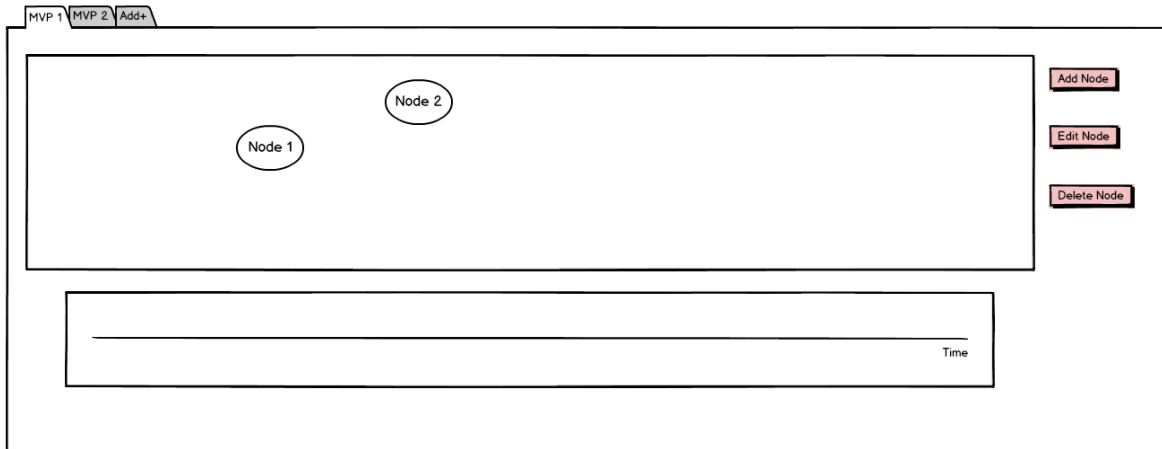


Figura 5. Desarrollo del MVP

En términos generales así se debería ver cada tarea, representada por un nodo en el área de trabajo.

3.2 Requisitos del Sistema.

Aclarando que nuestra herramienta pertenece a un proyecto mayor, existirán algunos requisitos del sistema que serán comunes con otras herramientas de la plataforma del programa ULA Startups y por ende ya están cubiertos en la plataforma. Por lo que nos enfocaremos sólo en los requisitos que debe cubrir GeMVP pero intentaremos mencionar todos los requisitos que intervienen en la herramienta.

3.2.1 Requisitos Funcionales

- Gestión de Usuarios (Miembros de la Startup y Supervisor).
- El sistema debe permitir al usuario pintar un grafo para representar el camino a seguir para la obtención de un MVP, de manera dinámica.
- El sistema debe permitir al usuario llenar cada actividad pintada en el grafo con información relacionada a las actividades a seguir.
- El sistema debe pintar automáticamente en la línea de tiempo, todas las actividades que se vayan registrando en el sistema.
- El sistema debe permitir editar el grafo.
- El sistema debe permitir editar cada actividad registrada en el grafo.
- El sistema debe permitir eliminar una actividad seleccionada del grafo.
- El sistema debe permitir la gestión de múltiples MVP.

3.2.2 Requisitos No Funcionales

- El sistema debe estar disponible en todo momento para los usuarios.
- El sistema debe tener una respuesta rápida para los usuarios.
- El sistema debe ser estable.
- El sistema debe ser dinámico, permitir la interacción de forma agradable con los usuarios.
- El sistema debe ser de uso simple.
- El sistema debe ser cómodo e intuitivo.
- El sistema debe ser escalable.

3.3 Modelo del Sistema

3.3.1 Modelado de Actores

GeMVP en su primera versión permitirá la interacción con dos tipos de actores:

Miembro Startup: Son los usuarios de la herramienta que pueden dibujar su grafo de tareas, editarlo, rellenar con información cada tarea, editar o eliminar las tareas, crear un nuevo MVP y consultar cada MVP que pertenezca a una Startup.

Evaluador: Son los usuarios que tienen acceso a la herramienta como observadores, en principio no pueden editar los MVP de las Startups, solo pueden revisar tanto el grafo de asignaciones como la línea de tiempo para revisar el estatus de la Startup, así como cuales asignaciones han sido cumplidas a tiempo, cuales se encuentran retrasadas y cuales están por empezar.

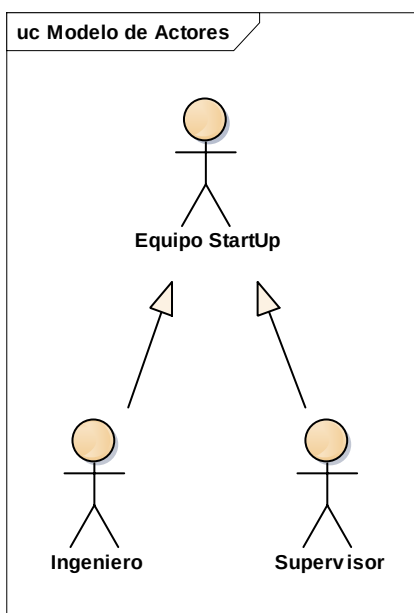


Figura 6. Modelo de Actores

3.3.2 Modelado de Asignaciones

Hitos: Se representan mediante los nodos del grafo de asignaciones, cada nodo por asignación, se pintan de manera dinámica, cada hito se puede conectar con otro hito mediante las aristas del grafo, y eso ocurre automáticamente cuando un usuario llena la información correspondiente de cada hito. La información mínima necesaria son: el nombre del hito, la descripción y la fecha de inicio.

Los hitos se pintan automáticamente en las dos áreas de trabajo, en el “Work Board” y en la Línea de Tiempo, pero sólo en el “Work Board” se pueden editar, en la línea de tiempo sólo se pueden observar.

Los hitos tienen colores que representan el estado en que se encuentre cada hito, amarillo si se encuentra por empezar, azul si se encuentra en progreso, verde si ya está finalizada y rojo si el hito se encuentra en estado retrasado.

3.3.3 Modelado de Áreas de Trabajo

Dentro de GeMVP se encuentran tres áreas de trabajo, las cuales se detallarán a continuación:

“Work Board”: Es el área de trabajo principal, la pizarra de trabajo donde cualquier miembro de la Startup puede pintar dinámicamente su grafo de asignaciones, y editar la posición de cada hito gracias a su componente de física el cual permite moverlos y arrastrarlos a lo largo de toda el área. Cuenta con un par de herramientas de visualización que permite hacer zoom dentro del grafo o movilizarse en el mismo.

Línea de Tiempo: Es el área de trabajo en donde tanto los miembros de la Startup y los evaluadores pueden revisar el estatus de cada hito de acuerdo a un orden cronológico, más no pueden editarlo.

Edición: Es una ventana dentro de la ventana, es el formulario que permite registrar toda la información correspondiente de cada hito, a su vez permite editar o eliminar los hitos y mediante esta ventana se pinta automáticamente la Línea de Tiempo.

3.3.4 Interacción

La interacción entre los actores, las áreas de trabajo y los hitos se realizan de la siguiente forma:

Un Miembro de la Startup puede:

❖ Crear un MVP

- Crear un Hito mediante:
 - Doble Click en el Work Board
 - Presionando Click en el botón añadir hito
- Editar un hito
 - Presionando click sobre el nodo
 - Llenando la información correspondiente del hito en la ventana de edición
 - Presionando el botón guardar en la ventana de edición
- Eliminar un hito
 - Presionando click sobre el nodo
 - Presionando el botón borrar en la ventana de edición
- Modificar la posición de los nodos
 - Seleccionar el nodo y arrastrarlo hasta la posición deseada
- Consultar la Línea de Tiempo:

- Hacer click sobre la línea de tiempo y arrastrar hacia la derecha o izquierda para consultar los hitos que están ordenados cronológicamente

Un evaluador puede:

- ❖ Consultar el Work Board ya sea:
 - Haciendo click y dejando presionado y moviéndose hacia cualquier dirección
 - Utilizando las herramientas de zoom
 - Presionando las flechas del teclado
- ❖ Consultar la línea de tiempo
 - Haciendo click sobre la línea de tiempo y arrastrar el cursor hacia la derecha o izquierda para consultar los hitos que están ordenados cronológicamente

3.4 Casos de Uso

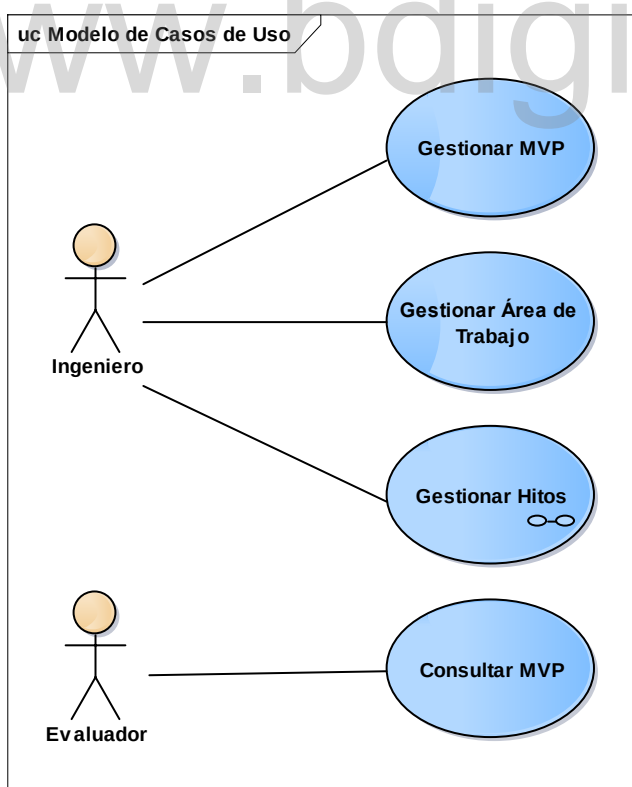


Figura 7. Modelo de Casos de Uso

3.4.1 Gestionar MVP

Nombre: Gestionar MVP
Descripción: Permite crear, editar y eliminar un MVP
Actores: Ingeniero
Precondiciones: El usuario debe pertenecer a una Startup
Flujo Normal: Crear: • El usuario introduce el nombre del MVP • El usuario presiona el botón de crear Editar: • El usuario selecciona un MVP de la lista de MVP • El usuario presiona el botón de editar Eliminar: • El usuario selecciona un MVP de la lista de MVP • El usuario presiona el botón de eliminar
Flujo Alternativo:
Poscondiciones: El usuario accede a la herramienta GeMVP

Tabla 1 Casos de Uso: Gestionar MVP

3.4.2 Gestionar Área de Trabajo

Nombre: Gestionar Área de Trabajo
Descripción: Permite visualizar, interactuar y editar las áreas de trabajo
Actores: Ingeniero
Precondiciones: Debe existir un MVP
Flujo Normal: Visualizar: • El usuario sin dejar de hacer click sobre un área de trabajo mueve el cursor hacia donde quiera visualizar Interactuar: • El usuario puede mover los nodos con la física permitida por la aplicación Editar: • El usuario puede editar la forma en que observa las áreas de trabajo
Flujo Alternativo:

Visualizar
El usuario utiliza las herramientas de visualización
Poscondiciones:
El usuario configuró las áreas de trabajo para mayor comodidad a la hora de trabajar

Tabla 2 Casos de Uso: Gestionar Área de Trabajo

3.4.3 Gestionar Hitos

Nombre:
Gestionar Hitos
Descripción:
Permite crear, editar y eliminar hitos del grafo de asignaciones
Actores:
Ingeniero
Precondiciones:
Debe existir un MVP
Flujo Normal:
Crear: - El usuario presiona el botón de crear un hito - El usuario introduce la información correspondiente de cada hito - El usuario presiona el botón de guardar información Editar: - El usuario selecciona un nodo del grafo - El usuario edita la información correspondiente - El usuario presiona el botón de guardar información Eliminar: - El usuario selecciona un nodo del grafo - El usuario presiona el botón de eliminar
Flujo Alternativo:
Crear - El usuario presiona doble click sobre el área de trabajo - El usuario introduce la información correspondiente de cada hito - El usuario presiona el botón de guardar información Editar - El usuario seleccionó un nodo equivocado - El usuario presiona el botón de cancelar Eliminar - El usuario seleccionó un nodo equivocado - El usuario presiona el botón de cancelar
Poscondiciones:
El usuario tiene un grafo de actividades

Tabla 3 Casos de Uso: Gestionar Hitos

3.4.4 Consultar MVP

Nombre: Consultar MVP
Descripción: Permite consultar un MVP para su evaluación
Actores: Evaluador
Precondiciones: Debe existir un MVP
Flujo Normal: El evaluador supervisa todo el grafo de actividades arrastrando el apuntador sobre el grafo
Flujo Alternativo: El evaluador supervisa todo el grafo de actividades utilizando las herramientas de movimiento y observación
Poscondiciones: El usuario conoce el estatus en que se encuentra el desarrollo del MVP

Tabla 4 Casos de Uso: Consultar MVP

www.bdigital.ula.ve

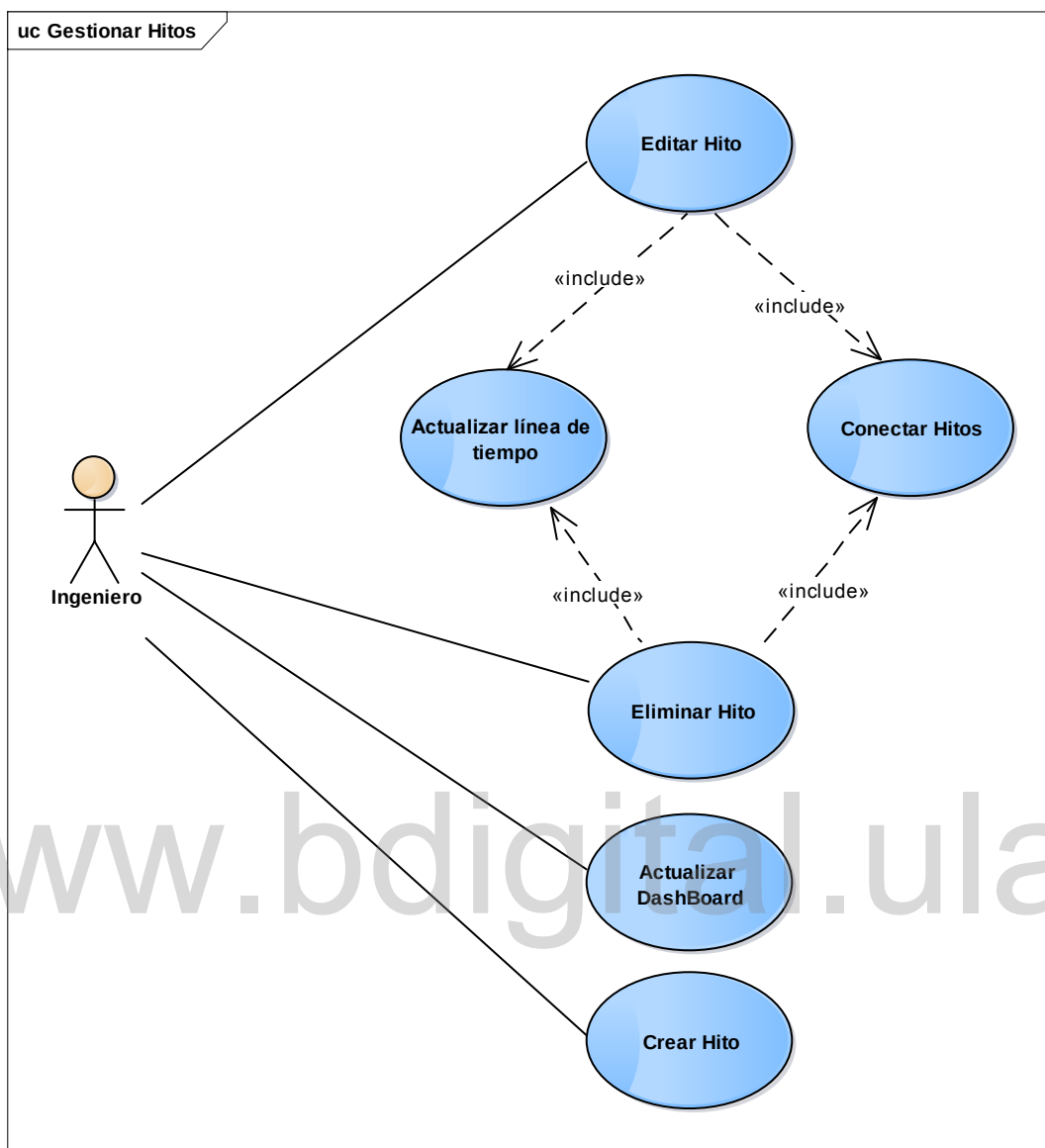


Figura 8. Modelo de Casos de Uso. Gestionar Hitos

3.4.5 Editar Hito

Nombre:
Editar Hito
Descripción:
Permite al usuario editar un hito del grafo de asignaciones
Actores:
Ingeniero
Precondiciones:
Debe existir un hito en el grafo de asignaciones
Flujo Normal:

• El usuario selecciona un hito del grafo de asignaciones • Se abre una ventana emergente con los datos del hito ya precargados • Se modifica el dato que se quiera editar • Se presiona el botón de guardar • Se activa el caso de uso Actualizar Línea de Tiempo • Se activa el caso de uso conectar hitos
Flujo Alternativo: • El usuario seleccionó el hito equivocado • Se debe presionar el botón de cancelar
Poscondiciones: El hito quedó actualizado

Tabla 5 Casos de Uso: Editar Hito

3.4.6 Eliminar Hito

Nombre: Eliminar Hito
Descripción: Permite al usuario eliminar un hito del grafo de asignaciones
Actores: Ingeniero
Precondiciones: Debe existir un hito en el grafo de asignaciones
Flujo Normal: • El usuario selecciona un hito del grafo de asignaciones • Se abre una ventana emergente con los datos del hito ya precargados • Se presiona el botón de eliminar • Se activa el caso de uso Actualizar Línea de Tiempo • Se activa el caso de uso conectar hitos
Flujo Alternativo: • El usuario seleccionó el hito equivocado • Se debe presionar el botón de cancelar
Poscondiciones: El hito quedó eliminado

Tabla 6 Casos de Uso: Eliminar Hito

3.4.7 Actualizar línea de tiempo

Nombre: Actualizar línea de tiempo
Descripción: El sistema actualiza la línea de tiempo automáticamente
Actores:

Se activa mediante otro caso de uso
Precondiciones: Fue activado por otro caso de uso
Flujo Normal: - El sistema consulta la base de datos - El sistema repinta la línea de tiempo
Flujo Alternativo: No existe flujo alternativo
Poscondiciones: La línea de tiempo quedó actualizada

Tabla 7 Casos de Uso: Actualizar línea de tiempo

3.4.8 Conectar Hitos

Nombre: Conectar Hitos
Descripción: El sistema conecta o desconecta los nodos del grafo automáticamente
Actores: Se activa mediante otro caso de uso
Precondiciones: Fue activado por otro caso de uso
Flujo Normal: - El sistema consulta la base de datos - El sistema repinta los nodos del grafo
Flujo Alternativo: No existe flujo alternativo
Poscondiciones: El grafo quedó actualizado con sus nodos conectados

Tabla 8 Casos de Uso: Conectar Hitos

3.4.9 Actualizar Dashboard

Nombre: Actualizar Dashboard
Descripción: El usuario puede distribuir la ubicación de los nodos del grafo a donde prefiera
Actores: Ingeniero
Precondiciones: Debe existir el grafo de actividades
Flujo Normal: El usuario selecciona el nodo que quiere mover y sin dejar de hacer click lo arrastra hasta su nueva posición

Flujo Alternativo: No existe flujo alternativo
Poscondiciones: El grafo quedó actualizado con sus nuevas posiciones

Tabla 9 Casos de Uso: Actualizar Dashboard

3.4.10 Crear Hito

Nombre: Crear Hito
Descripción: El usuario puede crear un hito de diversas maneras
Actores: Ingeniero
Precondiciones: Debe existir el MVP
Flujo Normal: - El usuario presiona el botón de crear hito - Se abre la ventana emergente - Se ingresan los datos correspondientes - Se presiona el botón de guardar
Flujo Alternativo: Se presiona doble click sobre el área de trabajo
Poscondiciones: Se creó un hito en el grafo de actividades

Tabla 10 Casos de Uso: Crear Hito

Capítulo 4

Justificación del Diseño y Arquitectura.

4.1 Justificación del Diseño del Sistema.

La plataforma web para el programa ULA Startups debe ser obligatoriamente multiusuario, de fácil acceso y disponible las 24 horas del día, con la menor cantidad de requisitos para su ejecución, esto en busca de que la puedan usar diferentes estudiantes o profesores en diferentes partes geográficas.

GeMVP como es una herramienta destinada a pertenecer a la plataforma web debe cumplir los mismos requisitos que el proyecto mayor a la que pertenece, por lo que se decidió usar HTML 5 como lenguaje de marcas para estructurar y mostrar la herramienta en la web.

Por la necesidad de ser multiusuario y que cada usuario puede aportar gran cantidad de información a la herramienta, así como el tamaño de los proyectos de cada Startup no es fijo, sino va creciendo con el tiempo, se decidió utilizar PHP 5.6.16 como lenguaje de programación y manejo de datos por su facilidad a la hora de manejar datos dinámicos y su diseño original orientado a la programación web.

Existe un Framework que ha tomado mucha fuerza en los últimos años llamado Laravel, el cual permite desarrollar aplicaciones y servicios web utilizando PHP 5 y sus versiones posteriores. Laravel fue el Framework utilizado debido a su filosofía de manejar un código simple y elegante, así como la facilidad que brinda a los nuevos usuarios del lenguaje PHP en su comprensión y manejo de datos, el cual fue nuestro caso. No estamos diciendo que sea la mejor opción posible haber elegido Laravel ni PHP, sino una elección basada en nuestros gustos particulares.

GeMVP tenía el reto de ser una herramienta agradable para el usuario, tenía que pintar de manera dinámica nodos de un grafo, así como hitos en una línea de tiempo. Se decidió afrontar este reto usando JavaScript por ser un lenguaje de programación que soporta la orientación a objetos y es muy dinámico, es un lenguaje mayormente usado para dibujar diversos tipos de formas, y darle movimiento a las mismas. Las dos áreas de trabajo, la línea de tiempo y el “Work Board” se pueden construir con este lenguaje sin mayores dificultades.

Pero la esencia de GeMVP la encontramos en poder dibujar nodos, moverlos a donde sea preferible, eliminarlos, todo esto dentro de una sola pantalla sin tener que hundir repetidamente click tras click. Para poder hacer todo esto se decidió apoyarse en una biblioteca de JavaScript conocida como VIS.JS, una biblioteca que no tiene mucho tiempo en uso, que se encuentra en

constante construcción pero que tiene ciertas características que nos sirven a la perfección, como lo es pintar nodos, agregarle física a los nodos, interacción con los nodos, pintar hitos en la línea de tiempo, eliminar nodos, formatear el grafo y muchas otras funcionalidades que seguramente servirán para escalar y mejorar esta herramienta en posibles trabajos futuros.

Para crear la interfaz gráfica se decidió seguir nuestro criterio de diseño buscando simplicidad para el uso de la herramienta, pocos clicks y que todo se encuentre al alcance dentro de una misma pantalla.

Finalmente, al haber mencionado la gran cantidad de datos y el dinamismo que se presenta en el uso de GeMVP, necesitamos una estructura donde guardar todos estos datos para su posterior uso, debido a esto, se decidió utilizar MySQL 10.1.9 como base de datos y Apache 2.4.17 como servidor por su alta popularidad en el mundo y su consistencia a la hora de manejar los datos.

4.2 Diagrama de despliegue.

Para usar el GeMVP, es necesario contar en el lado del cliente, con un navegador que soporte JavaScript y del lado del servidor, un servidor con capacidad para ejecutar código PHP como Apache y un manejador de base de datos como MySQL.

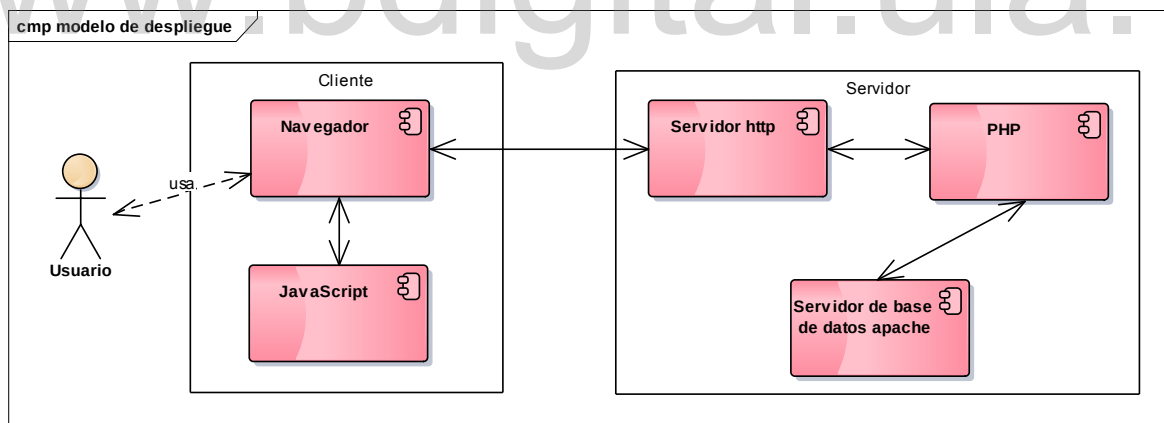


Figura 9. Diagrama de despliegue en donde se explica en qué orden viajan las peticiones de los clientes.

4.3 Arquitectura de software.

El patrón de arquitectura de software usado en la herramienta fue el MVC o Modelo-Vista-Controlador como consecuencia de haber usado el Framework Laravel. Ver figura 10.

Modelos: Son los encargados de conectarse con las tablas de la base de datos y manejar las inserciones o las solicitudes de datos.

Vistas: En nuestro caso es una sola vista que acumula todo lo explicado en la sección anterior, es lo que se le muestra al usuario por pantalla.

Controlador: Existen diversos controladores que se encargan de la programación Back-End, y son todos aquellos aspectos que el cliente no ve por pantalla pero que se ejecutan para traer los datos de cierta forma o poder pintar el grafo correspondiente.

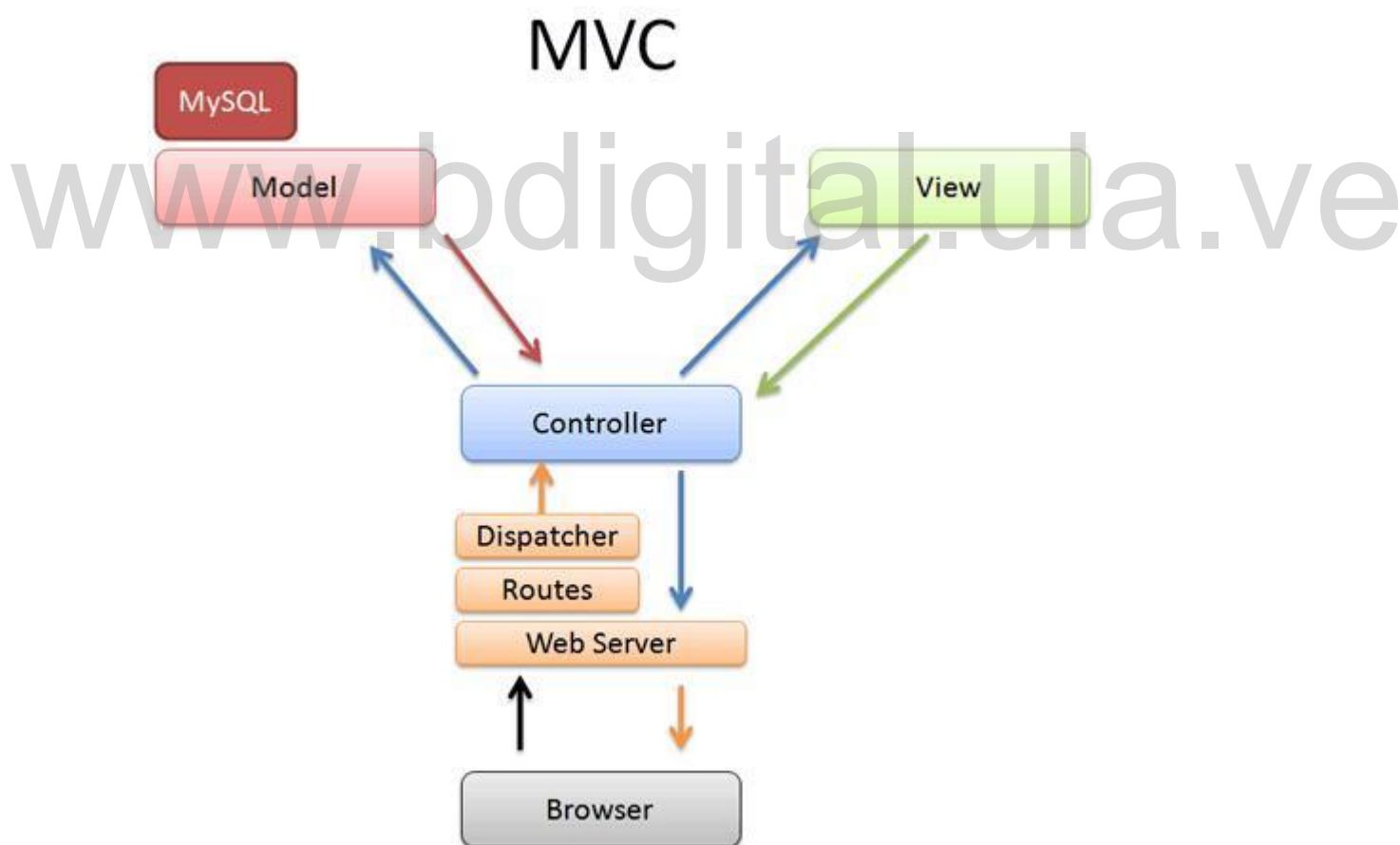


Figura 10. Arquitectura usada por Laravel, que a su vez es similar al usado por Codeigniter. [12]

El flujo de trabajo es el siguiente:

- La vista está esperando alguna interacción con el usuario
- Se envía la solicitud del usuario al controlador
- El controlador dependiendo de la solicitud le pide al modelo algún dato de la base de datos
- El modelo se conecta con la base de datos y devuelve el dato solicitado
- El controlador realiza las operaciones correspondientes con los datos que envió el modelo y le devuelve a la vista la respuesta de la solicitud del cliente
- La vista muestra los datos con el formato correspondiente
- La vista está esperando alguna interacción con el usuario

4.4 Modelo relacional de base de datos.

A continuación se muestra el esquema de base de datos relacional, que más adelante se explicará con más detalle.

- StartUp (id, name).
- MVP (id, name).
- Member (id, name).
- Hito (description, finish_date, id, name, seek_support, start_date).
- MemberHito (id, idHito, idMember).
- Arc (id, idFinal, idStart).
- Area (id, name).
- Priority (id, name).
- Hierarchy (id, name).
- Status (id, name).

Las relaciones entre las tablas (las claves foráneas) se expresan en el área de Apéndices.

Analizando las dependencias funcionales se puede determinar lo siguiente:

1. No existe en el modelo relacional atributos multi-evaluados y todas las tuplas tienen el mismo número de columnas, por lo tanto, está en primera forma normal.
2. Los atributos de todas las tablas dependen totalmente de la clave primaria y está en primera forma normal, por lo tanto está en segunda forma normal.
3. No existen dependencias transitivas en el modelo y está en segunda forma normal, por lo tanto, está en tercera forma normal.

4.5 Diagrama de clases.

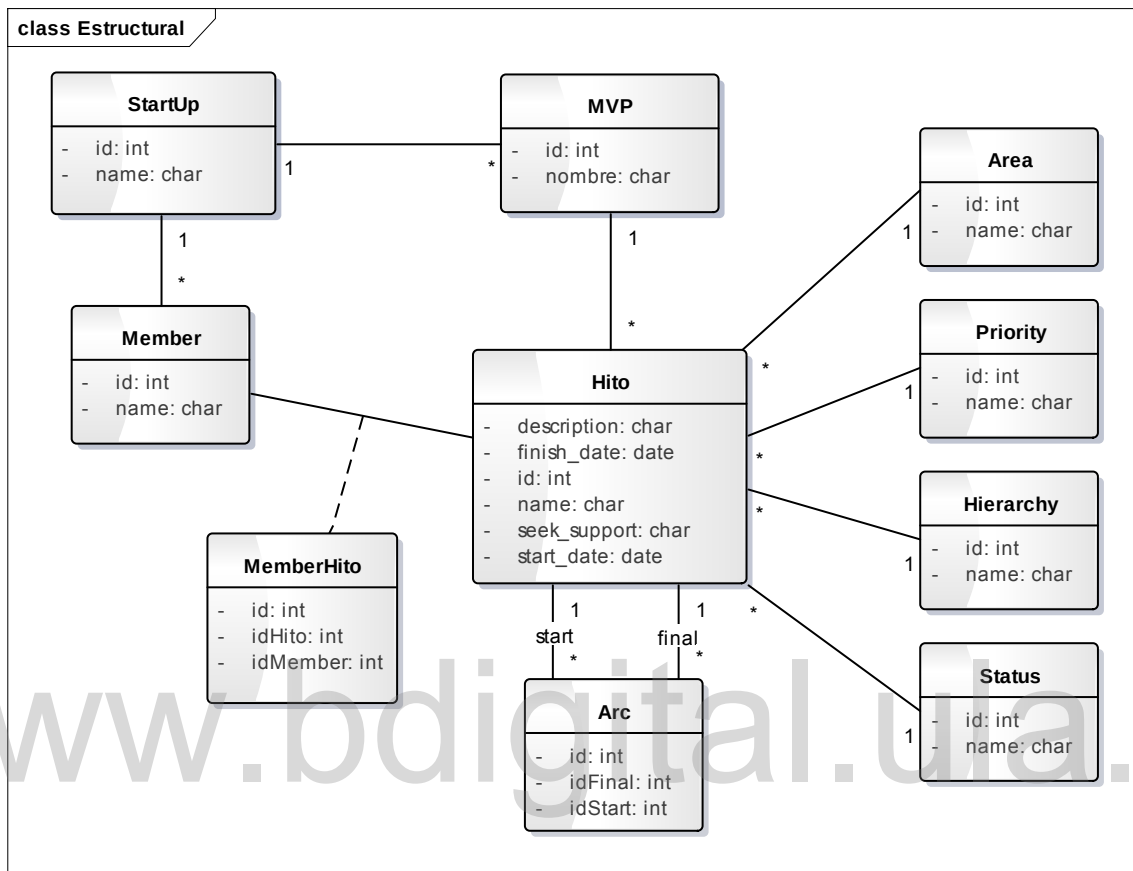


Figura 11. Diagrama de Clases

En principio este fue el modelo de base de datos usado para desarrollar la aplicación, posteriormente cabe recordar, que la herramienta tuvo que adaptarse a una plataforma más grande y una que otra tabla ya estaba creada en el proyecto mayor por lo que la herramienta tuvo que acoplarse a la base de datos mayor.

4.6 Diagrama de Paquetes

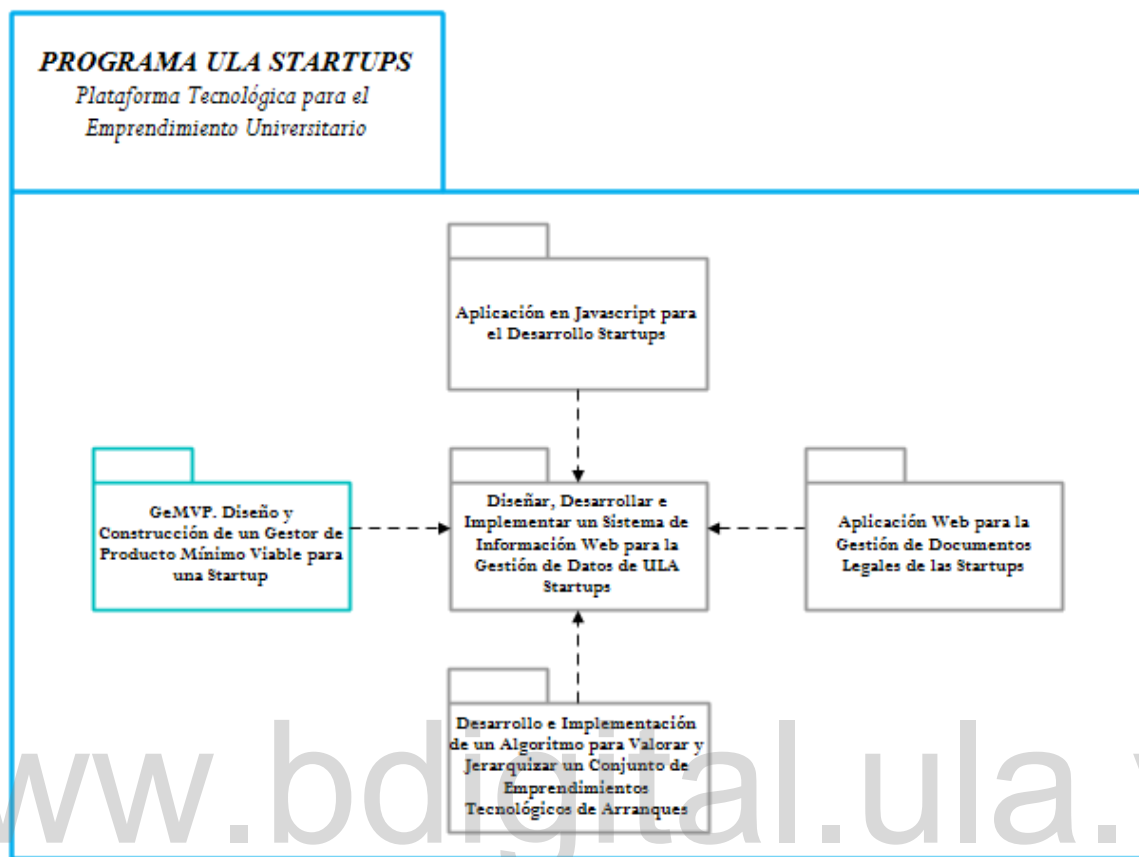


Figura 29. Diagrama de Paquetes de la integración de las herramientas del Programa ULA Startups.

Capítulo 5

Desarrollo de la Aplicación

5.1 Descripción de la Metodología:

A nivel organizacional de la plataforma tecnológica del programa ULA STARTUPS, se utilizó la estrategia R.A.I.S (Reproducción del ambiente de innovación del salón de clase), que permitió organizar el grupo de trabajo como una Startup, cuyos miembros son los responsables de la construcción de cada herramienta bajo la supervisión del tutor, cuyo rol es ser el jefe ejecutivo. De esta manera, se permite establecer responsabilidades necesarias para lograr los objetivos planteados y a su vez estimular a los integrantes del equipo a sumar esfuerzos y competencias para desarrollar juntos la plataforma tecnológica del programa ULA STARTUPS.

Para la integración de las herramientas se hacían reuniones semanales teniendo en cuenta los siguientes principios, cero tolerancia a excusas, informe de que se logró durante la semana, informe de que problemas se presentaron y planificación del trabajo a realizar para la próxima reunión, principios fundamentales de la estrategia R.A.I.S.

Siempre llevando un control de la información y documentación del progreso utilizando medios digitales como Trello, WhatsApp entre otros.

A nivel particular de la herramienta en sí, GeMVP fue desarrollada bajo la metodología del modelo de desarrollo en Espiral propuesto por Barry Boehm, principalmente por su alto crecimiento mientras disminuye el grado de riesgo en cada entrega semanal.

5.2 Desarrollo de GeMVP:

Para poder manejar versiones de la herramienta y hacer pruebas con cada una de ellas, se utilizó un repositorio web llamado GIT, el cual nos permitía descargar versiones en modo lectura o en modo de edición, ya sea para probar cada versión con sus nuevas funcionalidades o para agregarle nuevas a las versiones anteriores.

Ahora se procede a explicar el procedimiento de construcción de GeMVP:

En primer lugar se instaló el Framework Laravel, desde su sitio web oficial <https://laravel.com>, posterior a esto se instaló un gestor de dependencias de proyectos el cual es obligatorio para

programar en PHP usando este Framework y permite mantener actualizados todos los paquetes y dependencias que vayamos a usar, este gestor de dependencias de proyectos se llama Composer.

De las primeras acciones realizadas, fue la de crear la base de datos, para eso se utilizó la herramienta de php llamada PHPMYAdmin, que nos permitió manejar y administrar la base de datos desde un navegador web.

Para poder configurar y emparejar la base de datos con nuestro proyecto en Laravel, se modificó el archivo “.env” dentro del proyecto, asignándole los valores correspondientes al usuario y contraseña para la base de datos, así como el Host donde está ubicada y el nombre de la misma, este paso es el primero que se debe seguir para empezar cualquier proyecto usando Laravel ya que permite hacer la conexión con la base de datos indicada sin tener ningún inconveniente y te facilitará posteriormente las conexiones ahorrando muchas líneas de código.

GeMVP tiene una arquitectura de software MVC que ya explicamos en secciones anteriores, de este modelo podemos decir que verdaderamente se usó la vista y el controlador, ya que el modelo quedó gestionado directamente desde el controlador haciendo conexiones con la base de datos con una sola línea de código como por ejemplo

```
$variable = DB::table('miTabla')->get();
```

Lo primero en construir fue la vista, siguiendo el objetivo de que fuese una sola vista para toda la herramienta, “start.blade.php” se convirtió en la cara de la herramienta, fue construida usando HTML 5, CSS 3, JavaScript y PHP, con un organizador de tamaño Responsive, que permite visualizar con el tamaño correcto la vista dependiendo del dispositivo en el que se vea. Se construyeron las tres áreas de trabajo en principio vacías, y la ventana de edición fue construida mediante un Modal, que al iniciar la aplicación se encuentra dentro de la pantalla pero no se puede ver, sino hasta que se activen o se realicen las acciones correspondientes es cuando se podrá visualizar la ventana de edición.

Para darle vida a la vista de GeMVP, usamos la biblioteca de JavaScript VIS.JS, la cual fue descargada desde su sitio web oficial <https://visjs.org>, y se instaló en el proyecto Laravel dentro del directorio “{raiz}/public/dist”. Una vez teniendo la biblioteca activa, se crearon dos estructuras dentro de cada área de trabajo, la primera correspondiente a una estructura de Network dentro del Work Board y la otra una estructura Timeline dentro del área línea de tiempo.

En principio con lo descrito anteriormente tenemos todo el esqueleto de GeMVP, pero faltan todos los datos, para el manejo de datos se crearon distintos controladores como lo son:

- HitosController.php
- JsonController.php

HitosController.php: es el controlador encargado de manejar todos los datos del grafo y la línea de tiempo, es el controlador que se conecta con la base de datos para guardar, actualizar, eliminar y hasta extraer los datos correspondientes para enviarlos a la vista. Pero se presentó un problema,

los datos que vienen de la base de datos el controlador los manejaba con formato PHP, y en la vista estamos utilizando una biblioteca JavaScript, es decir los datos no eran compatibles para mostrar y pintar los nodos, por ello se creó el controlador JsonController.php.

JsonController.php: es el controlador que sirve para formatear los datos que vienen de HitosController.php en un lenguaje “universal” para ambos lenguajes como es el formato JSON, es decir, actuó como puente entre la vista y su controlador principal, ya que permitió formatear los datos que venían en lenguaje PHP hacia JSON y devolvía los datos a la vista con formato JavaScript, permitiendo pintar tanto el grafo como la línea de tiempo en la vista.

Este controlador sólo se usaba cuando se devolvían datos del controlador a la vista, debido a que en sentido contrario, de la vista al controlador los datos se pasaban directamente en formato PHP.

En una de las versiones finales de la herramienta, se incorporó un par de herramientas para la visualización del grafo, esto se hizo utilizando JavaScript, constaba de unas “flechas de teclado virtual” y dos botones de zoom “+ y -”.

La esencia de toda la herramienta se encuentra en su simplicidad, ya que para crear un nodo, tienes diversas formas de hacerlo, como por ejemplo, dando doble click dentro del Work Board, así como presionando click sobre el botón añadir nodo, dejándole al usuario la decisión de usar la opción que más le agrade.

Para rellenar los nodos con información, sólo se debe hacer click encima de un nodo y aparecerá automáticamente la ventana de edición en un recuadro en la derecha inferior de la pantalla, el cual no es más que un formulario con los botones respectivos guardar, cerrar o eliminar.

La configuración de la física tanto de la línea de tiempo como del grafo se asigna en la opción “Network/options/interaction” y “Timeline/options/interaction” asignándole el valor “True”.

5.3 Pruebas del Sistema

Durante el desarrollo del GeMVP y queriendo emular el ambiente de trabajo que se maneja en la realidad de una compañía tecnológica, había una reunión semanal para reportar avances de la aplicación, en donde podemos decir que se entregaba una versión diferente semana tras semana, en donde pudimos probar la herramienta con diferentes tipos de usuarios.

En principio, compañeros de la escuela de ingeniería de sistemas fueron los encargados de probar las primeras versiones de la aplicación y dejar sus opiniones sobre que mejorar y contar sus experiencias tanto positivas como negativas al usar GeMVP.

De esas primeras pruebas fue en donde se determinó la importancia de tener toda la herramienta en una sola pantalla, buscando la simplicidad y cierto tipo de elegancia evitando los múltiples clicks que para la mayoría de los usuarios representaba una gran molestia.

A su vez, descubrimos la necesidad de tener varias formas de ingresar datos, ya que en algunos casos el botón de agregar era de extrema necesidad, pero en otros, lo novedoso del doble click encima del área de trabajo era ideal.

Cuando la herramienta se encontraba en su fase “final”, fue cuando se probó con usuarios reales, emprendedores de la ciudad de Mérida con ideas de productos y servicios novedosos, pero sin nada concreto. Ellos fueron los encargados de hacer la planificación correspondiente de acuerdo a cada caso en específico para conseguir el MVP de cada proyecto, usando GeMVP como el gestor de desarrollo. Se dieron como tiempo de desarrollo un total de 4 semanas por proyecto, en donde usaron la herramienta en su totalidad, arrojando los siguientes resultados:

- La herramienta se comportó de manera estable durante todo el proceso
- GeMVP fue sencilla de usar, de rápido aprendizaje para los nuevos usuarios
- GeMVP contribuyó de manera positiva en la organización de ideas para la consecución de un producto mínimo viable
- La herramienta mostró una enorme capacidad de escalabilidad, dejando muy claro las áreas en que puede crecer
- La herramienta permitió a los evaluadores poder supervisar constantemente el crecimiento y desarrollo de cada MVP

Capítulo 6

Manual de usuario

GeMVP es una herramienta que ayuda a las Startups a planificar los pasos a seguir para concebir un producto mínimo viable.

6.1 Manual de usuario.

Pantalla inicial para cualquier usuario:

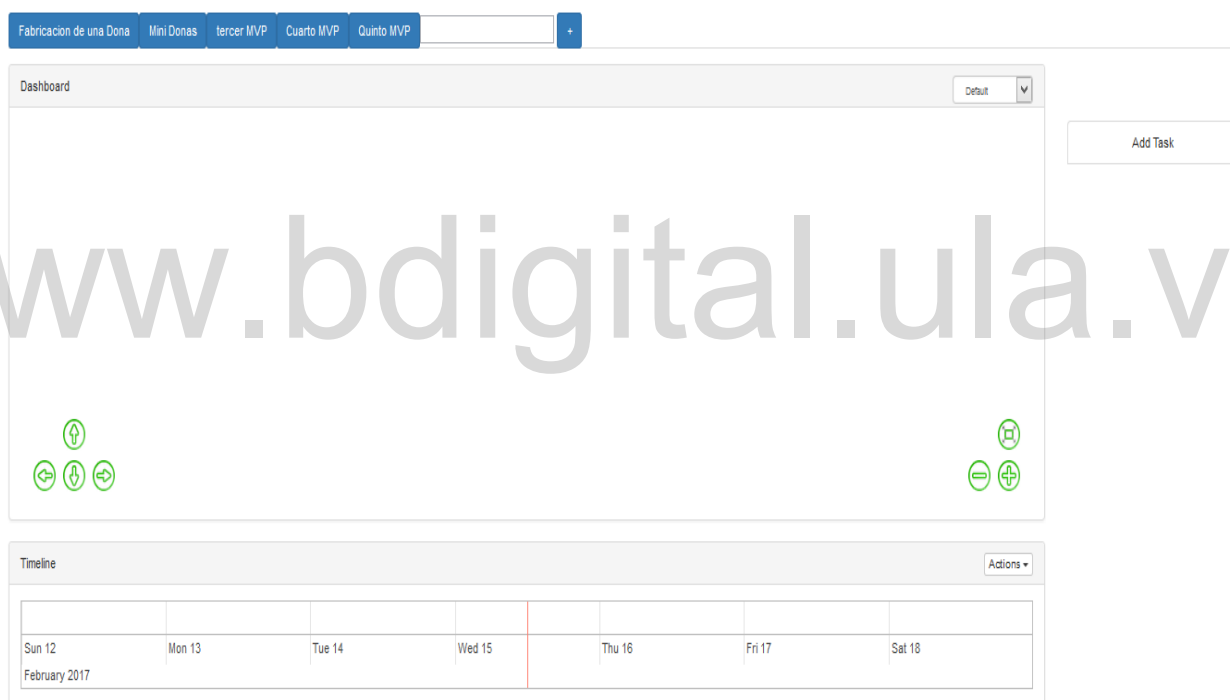


Figura 12. Pantalla inicial

En la figura 12 podemos apreciar las dos áreas de trabajo de la herramienta como lo son el DashBoard y la Línea de tiempo, así como el botón de agregar hito y los botones de visualización del grafo de tareas. Es la única pantalla que se desea para toda la aplicación.

También podemos notar que la línea de tiempo marca con una línea roja el día actual en que se encuentre el usuario.

Como paso uno podemos agregar un hito de dos maneras.

La primera forma es presionando click en el botón de agregar hito.

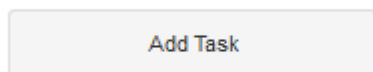


Figura 13. Botón de agregar hito

La segunda forma es haciendo doble click dentro del DashBoard.



Figura 14. DashBoard

De cualquiera de las dos formas obtendremos el mismo resultado.



Figura 15. Nodo Nuevo

En donde podemos visualizar el nuevo nodo creado.

Las características del nodo son: tiene un color amarillo por defecto ya que su estado por defecto es por empezar, y el nombre del nodo es "I am new".



Figura 16. Nodo nuevo detallado

Ahora se presenta una de las funcionalidades más importantes de la herramienta, al pasar el cursor por encima del nodo, se podrá visualizar su nombre y su descripción

www.bdigital.ula.ve

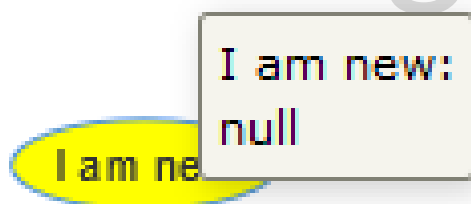


Figura 17. Visualización del nombre y descripción de cada nodo

Como el nodo está recién creado su valor por defecto para la descripción será Null

Seguidamente se presiona click sobre el nodo y se abre una ventana emergente.

The 'Add Task' modal window contains the following fields and controls:

- Title:** Text input field containing 'I am new'.
- ID:** Text input field containing '31'.
- Description:** Text area with placeholder 'Description of the Task'.
- Area:** Dropdown menu with 'Salud' selected.
- Priority:** Dropdown menu with 'Low' selected.
- Hierarchy:** Dropdown menu with '1' selected.
- Previous:** Dropdown menu with 'Vacío' selected.
- Start Date:** Text input field containing '2016-10-20'.
- Finish Date:** Text input field containing '2016-10-20'.
- Member List:**
 - ☐ anny donas
 - ☐ jose donas
- Status:** Dropdown menu with 'To Start' selected.
- Seek Support:** Text input field.
- Buttons:** 'Delete' (blue), 'Save' (blue), and 'Close' (white).

Figura 18. Ventana emergente para gestionar nodos

Esta es la ventana para gestionar los nodos, dentro de ella podemos editar un nodo, así como eliminarlo.

Para editar la información de un nodo se debe rellenar con la información correspondiente cada espacio de la ventana emergente.

Add Task

Title: Esta es la tarea 1 **ID** 31

Description: Esta es una descripcion de prueba

Area: Tecnologia **Priority:** Hight **Hierarchy:** 5 **Previous:** Vacio

Start Date: 2017-02-15 **Finish Date:** 2017-02-17

Status: To Start **Seek Support**

Member List

- ☒ anny donas
- ☐ jose donas

Delete **Save** **Close**

Figura 19. Editar nodo

Para posteriormente presionar el botón de guardar



Figura 20. Botón de guardar

Una vez presionemos el botón de guardar tendremos el nodo con su información correspondiente

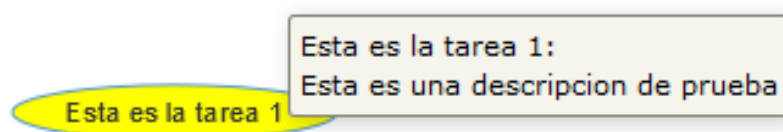


Figura 21. Nodo con su información correspondiente

A su vez la línea de tiempo se actualizó automáticamente con las fechas que ingresamos en el nodo

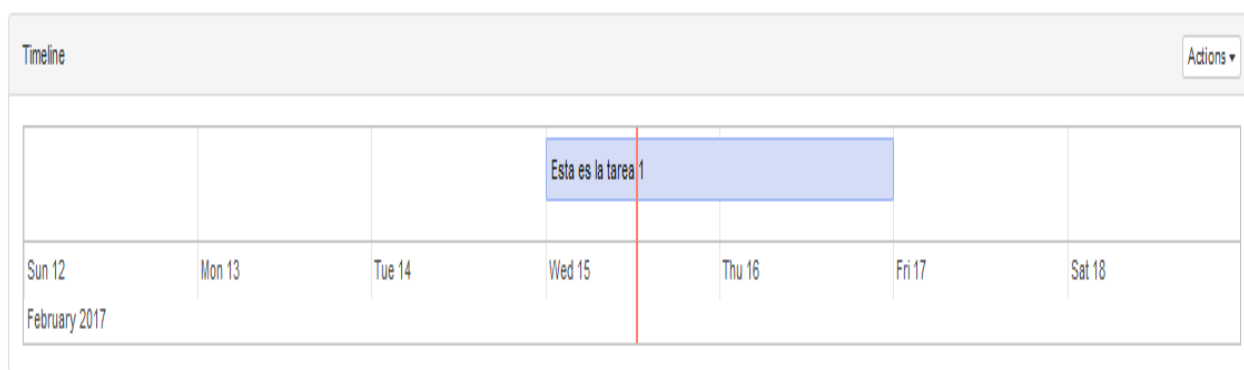


Figura 22. Línea de tiempo actualizada

Ahora, al generar más de un hito aparecen los arcos del grafo, se asigna en la casilla “previo”

Previous:

Figura 23. Casilla “Previo”

En la cual seleccionamos el hito previo de acuerdo a su ID en este caso de ejemplo el ID 31 para obtener el grafo de la siguiente manera



Figura 24. Grafo de actividades

En donde podemos notar que el grafo es un dígrafo ya que la prelación de actividades se ve representada mediante un arco dirigido. Ambos hitos se encuentran en color amarillo debido a su estado por empezar.

Automáticamente la línea de tiempo se va actualizando.

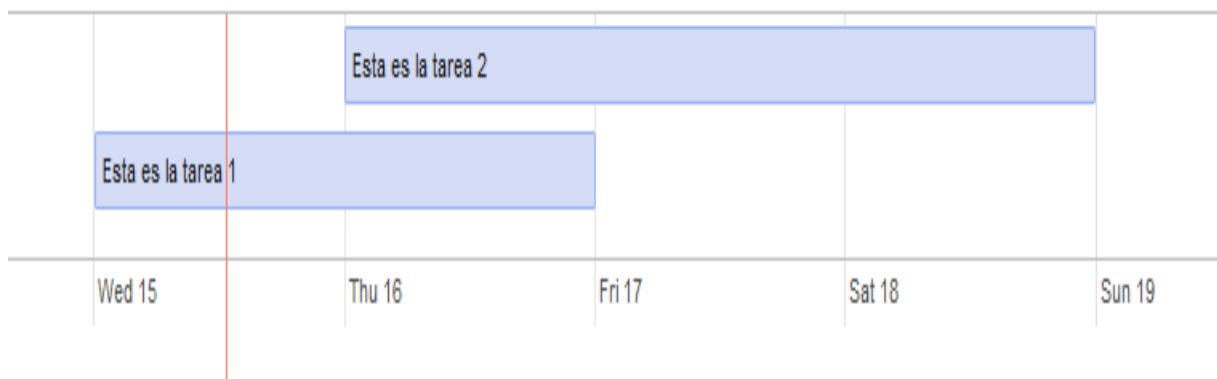


Figura 25. Línea de tiempo actualizada con dos hitos.

Una de las cosas que se deben mencionar acá, es que la visualización de la línea de tiempo es dinámica, puedes recorrer toda la línea de tiempo en su total extensión hacia ambos lados, sólo con el cursor sin dejar de hacer click y corriéndolo hacia derecha o izquierda según sea el deseo del usuario.

Esta es la lógica de la herramienta, se pueden crear un gran número de hitos, la herramienta fue probada hasta 100 sin presentar ningún problema.

Nos falta presentar las herramientas de visualización que no dependen de correr el cursor haciendo click, sino utilizando las flechas del teclado o el cursor en las flechas virtuales

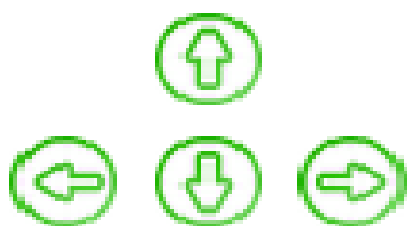


Figura 26. Flechas virtuales para visualización del grafo

Estas flechas se pueden usar tanto con las flechas del teclado como con el cursor haciendo click sobre las flechas.

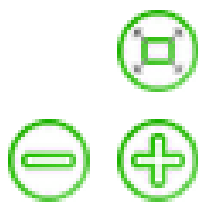


Figura 27. Herramienta de visualización botones de zoom

Estos botones de zoom se pueden usar tanto con los botones + y – del teclado o con el cursor haciendo click sobre ellos.

Finalmente se muestra una vista de toda la herramienta con diversos hitos en distintos estados.

www.bdigital.ula.ve

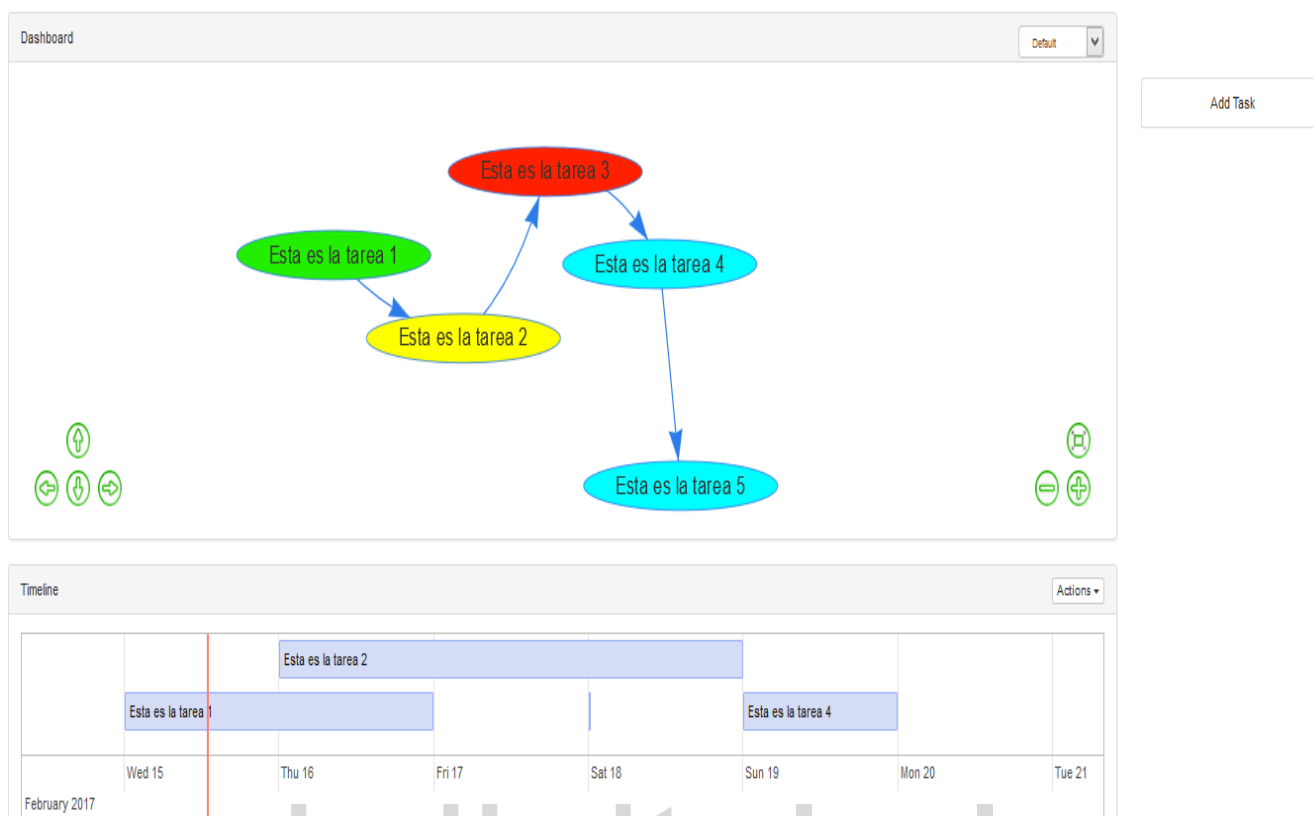


Figura 28. Vista general

Capítulo 7

Conclusiones y Recomendaciones

7.1 Conclusiones.

Después de haber tomado un curso completo de lo que significa el movimiento StartUp en el mundo, conocer la importancia del emprendimiento tecnológico, darnos cuenta que la evolución de la tecnología crece a nivel exponencial, y que la gran mayoría de los emprendedores fracasan, se identificaron las grandes necesidades que tienen los emprendedores para desarrollar sus ideas y productos novedosos, puntualmente a la hora de concebir un producto mínimo viable que de pie a dar a conocer todas aquellas ideas que parecen brillantes pero que sin llevar a cabo sólo quedan en ideas. Se diseñó, desarrolló y probó un Gestor de Productos Mínimo Viable, GeMVP, basada en una arquitectura web cliente servidor, que permita a todos aquellos emprendedores planificar y seguir el desarrollo de sus ideas, hasta llevarlos a la concepción de su MVP, contribuyendo así, de manera positiva con las Startups y cumpliendo con el objetivo principal de este trabajo.

Mientras se iba implementando cada versión de la herramienta, se validó cada una de sus funcionalidades, empezando por obtener un diseño exitoso y prometedor, siguiendo con la obtención de una herramienta funcional que podía gestionar todo el desarrollo de un producto mínimo viable y mostrando un correcto desempeño durante las distintas pruebas realizadas.

De esta manera se concluye que GeMVP contribuye de manera positiva con el desarrollo de las ideas de las Startups, planificando ordenadamente los pasos a seguir para la obtención de un Producto Mínimo Viable.

Dentro del contexto en que todos tienen ideas, pero pocos las llevan a cabo, aparece GeMVP como una herramienta a usar, dentro de un ambiente guiado como una aceleradora de Startups que les permitirá tanto a emprendedores como a evaluadores, formar equipos tecnológicos para darle forma a todas aquellas ideas y generar productos, este es el valor agregado de la herramienta.

7.2 Recomendaciones

Todas las pruebas a las que fueron sometidas las distintas versiones de GeMVP, sirvieron para puntualizar el gran nivel de escalabilidad que tiene esta herramienta, esperando que sirva como base para futuros desarrollos y poder explotar las incontables posibilidades que tiene GeMVP de convertirse en una herramienta mucho más completa y pionera en el mundo Startup.

Algunas de las recomendaciones recibidas de parte de los usuarios de GeMVP fueron:

- Poder sacar estadísticas de cada MVP
- Utilizar la teoría de grafos para optimizar el desarrollo de un MVP
- Desarrollar la herramienta dirigida a dispositivos móviles
- Que se activen las funcionalidades de manera táctil (para dispositivos móviles)
- Que se puedan insertar hitos desde la línea de tiempo
- Que genere una comunicación instantánea con el usuario encargado a cumplir un hito en específico
- Que permita imprimir el grafo de tareas directamente
- Poder manejar múltiples MVP dentro de la misma pantalla de la herramienta

Insistimos en que GeMVP tiene mucho potencial para crecer como herramienta en sí y se recomienda enfáticamente para posteriores trabajos, poder desarrollar la versión 2.0 de esta herramienta, cubriendo todos los requisitos que se generaron gracias a las sugerencias mencionadas anteriormente.

Bibliografía

- Peris R, (2014). Start-Ups Tecnológicas: El Reto del Conocimiento Global. Proyecto de grado, Universidad Pontificia Comillas, Madrid.
- Gutiérrez I, (2013). Creación de un emprendimiento de software utilizando metodología Lean Startup. Proyecto para optar al título de Ingeniero Informático, Universidad de Belgrano, Buenos Aires,
- Muñoz J, (2012). Diseño de un programa de gestión para la evaluación y rediseño continuo de un Startup. Proyecto para optar al título de Ingeniero Civil Industrial, Universidad de Chile, Santiago.
- Fino D, (2013). Innovación en modelos de negocio: Metodología Lean Canvas en una Startup de base tecnológica. Proyecto para optar al título de Especialista en Alta Gerencia, Universidad Militar Nueva Granada, Bogotá
- [1]. <https://es.shopify.com/blog/14934065-hablemos-sobre-startups-que-son-sus-caracteristicas-e-importancia>
- [2]. <http://www.desarrolloweb.com/articulos/que-son-gestores-proyectos.html>
- [3]. <http://tareasmmd.blogspot.com/2012/07/digrafos.html>
- [4]. <http://www.salutip.com/2012/04/que-es-una-linea-del-tiempo.html>
- Por qué fracasan las pymes. <http://www.gestiopolis.com/por-que-fracasan-las-pymes/>. Claudio Soriano (2005)
- [5]. https://es.wikipedia.org/wiki/Aplicaci%C3%B3n_web
- [6]. <https://es.wikipedia.org/wiki/Framework>
- [7]. https://es.wikipedia.org/wiki/Lenguaje_de_programaci%C3%B3n
- [8]. https://es.wikipedia.org/wiki/Base_de_datos
- [9]. https://es.wikipedia.org/wiki/Aceleradora_de_startups
- [10]. <https://styde.net/laragon-un-entorno-de-desarrollo-para-laravel-en-windows/>
- [11]. <https://es.wikipedia.org/wiki/Laravel>
- Santos García, G., & Sánchez García, J. I. (1997). HTML iniciación y referencia. Madrid: McGRAW-HILL
- [12]. <https://desarrolloweb.com/manuales/manual-codeigniter.html>
- [13]. <http://visjs.org/>

Apéndices

Apéndice. Glosario y definiciones

Definiciones técnicas.

El sistema a desarrollar funciona bajo una arquitectura Cliente-Servidor, la cual consiste en al menos dos aplicaciones en donde la aplicación cliente realiza peticiones a la aplicación servidor. Por su misma naturaleza esta arquitectura es la predilecta para sistemas multiusuarios la cual generalmente provee servicios en una red de computadores.

A continuación se explican los conceptos necesarios para entender los detalles del desarrollo y funcionamiento del sistema a elaborar.

HTML

(Acrónimo de HyperText Markup Language o Lenguaje de Marcado de Hipertexto), es un lenguaje de marcado basado en XML que especifica la organización lógica de un documento. Es usado principalmente para exponer información en los sitios y páginas WEB (Santos García & Sánchez García, 1997).

Navegador Web

Son aplicaciones de escritorio capaces de interpretar y presentar contenidos en hipertexto al usuario permitiéndole interactuar con él. El nombre de navegador hace referencia a que los contenidos en hipertexto normalmente están cargados con hipervínculos que permiten enlazar otros contenidos que a su vez también tienen otros hipervínculos. Al pasear por los mismos se podría decir que se está navegando en el mar de la información; esa es la razón de su nombre.

Existen muchas aplicaciones de este tipo, disponibles para casi todas las plataformas disponibles, entre los cuales tenemos Firefox, Safari, Opera, Internet Explorer entre otros (Santos García & Sánchez García, 1997).

¿Qué es un Framework?

Es una estructura conceptual y tecnológica de soporte definido, normalmente con artefactos o módulos concretos de software, que puede servir de base para la organización y desarrollo de software. Típicamente, puede incluir soporte de programas, bibliotecas, y un lenguaje interpretado, entre otras herramientas, para así ayudar a desarrollar y unir los diferentes componentes de un proyecto. [6]

¿Qué es un lenguaje de programación?

Es un lenguaje formal diseñado para realizar procesos que pueden ser llevados a cabo por máquinas como las computadoras. Pueden usarse para crear programas que controlen el comportamiento físico y lógico de una máquina, para expresar algoritmos con precisión, o como modo de comunicación humana. [7]

¿Qué es una base de datos?

Es un conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso. En este sentido; una biblioteca puede considerarse una base de datos compuesta en su mayoría por documentos y textos impresos en papel e indexados para su consulta. Actualmente, y debido al desarrollo tecnológico de campos como la informática y la electrónica, la mayoría de las bases de datos están en formato digital, siendo este un componente electrónico, por tanto se ha desarrollado y se ofrece un amplio rango de soluciones al problema del almacenamiento de datos. [8]

PHP

(Acrónimo de PHP Hypertext Preprocessor o Preprocesador de Hipertexto PHP) Es un lenguaje de programación interpretado multipropósito y multiparadigma de código abierto, predilecto para la creación de sistemas web gracias a su característica de poder ser incrustado dentro del código HTML. Dentro de una arquitectura web Cliente-Servidor se encuentra corriendo en el servidor. Se distribuye bajo licencia GPL (Grupo PHP, 2012).

MVC (Modelo Vista Controlador)

Es un patrón de arquitectura de software que consta de las siguientes 3 capas:

- Modelos: Es la representación de una abstracción en forma de datos u objetos.
- Vistas: Es la representación pictórica de los modelos, cada modelo puede tener una o varias vistas. Esta puede realizar operaciones sobre los modelos.
- Controladores: Representa la lógica de negocio, es el enlace del usuario con el sistema permitiéndole a este interactuar con el resto de las capas

(Reenskaug, 1979)

Funcionalmente, un cliente solicita información al controlador, éste puede hacer uso del modelo para consultarla (si es necesario), luego se la presenta al solicitante por medio de la vista.

Este procedimiento se ilustra en la Figura 10.

JavaScript

Es un lenguaje de programación interpretado normalmente por los navegadores web; por lo tanto, puede correr del lado del cliente. Es usado para darle capacidades dinámicas a los contenidos HTML, como por ejemplo, menús de contenido, ventanas emergentes, validaciones de datos introducidos por el usuario, entre muchas otras funcionalidades (Santos García & Sánchez García, 1997).

SQL

(Acrónimo de Structured Query Language o lenguaje de consultas estructurado). Es un lenguaje desarrollado por IBM Research a modo de interfaz que permite la consulta, inserción y manipulación de los datos almacenados en una base de datos relacional. Es usado por varios servidores de bases de datos, como por ejemplo MariaDB, PostgreSQL, MySQL, SQLite entre otros (Elmasri & Navathe, 2007).

MySQL

Es un sistema de gestión de base de datos relacional multiusuario propiedad de Oracle, se distribuye bajo doble licencia siendo una de ellas libre, utiliza SQL para el manejo de los datos y posee interfaces para conectarse desde múltiples lenguajes incluyendo PHP (Oracle Corporation, 2011).

Laragon

Laragon es una suite de desarrollo para PHP que funciona sobre Windows diseñado especialmente para trabajar con Laravel. Similar a otras herramientas como Xampp o Wampp, Laragon nos permite crear un entorno de desarrollo con estas características: Cmder (Consola para Windows). [10]

Laravel

Laravel es un framework de código abierto para desarrollar aplicaciones y servicios web con PHP 5 y PHP 7. Su filosofía es desarrollar código PHP de forma elegante y simple, evitando el "código espagueti". Fue creado en 2011 y tiene una gran influencia de frameworks como Ruby on Rails, Sinatra y ASP.NET MVC. [11]

Vis.js

Vis.js es una biblioteca dinámica de javascript, que permite manejar grandes cantidades de datos, de manera fácil, dinámica y que es capaz de trabajar con esos datos darle forma e interactuar con ellos. [13].

Código Fuente de la aplicación

El proyecto GeMVP fue alojado en los repositorios con control de versiones GitHub. Siendo un código en un repositorio privado descargable sólo por los desarrolladores.

Registro de cambios.

Todos los cambios son registrados de manera que pueden ser revisados y auditados en incluso es posible recuperar versiones anteriores gracias al sistema de control de versiones. Todo dentro de GitHub con un perfil privado para los desarrolladores.

Estructura de la base de datos

A continuación se muestra la descripción de cada una de las tablas utilizadas en GeMVP.

Tabla: StartUp

Almacena los datos correspondientes a cada StartUp que pertenezca al programa ULA Startups.

Nombre	Tipo	Descripción
Id	Int unsigned	Clave Primaria
Name	Char	Nombre de la Startup

Tabla 11. Definición de la tabla StartUp.

Tabla: MVP

Almacena los datos correspondientes a cada MVP que pertenezca a cada Startup

Nombre	Tipo	Descripción
Id	Int unsigned	Clave Primaria
Nombre	Char	Nombre del MVP
idMVP	Int unsigned	Id de la Startup a la cual pertenece cada MVP Clave Foránea

Tabla 12. Definición de la tabla MVP.

Tabla: Member

Almacena los datos correspondientes a cada miembro de las Startups

Nombre	Tipo	Descripción
Id	Int unsigned	Clave Primaria
Name	Char	Nombre de cada miembro de la Startup
idStartup	Int unsigned	Id de la Startup a la cual pertenece cada miembro Clave Foránea

Tabla 13. Definición de la tabla Member.

Tabla: Hito

Almacena los datos correspondientes a cada hito que pertenece a un MVP

Nombre	Tipo	Descripción
Id	Int unsigned	Clave Primaria
Name	Char	Nombre de cada hito
idMVP	Int unsigned	Id del MVP al cual pertenece cada hito Clave Foránea
Description	Char	Descripción de cada hito
Seek_support	Char	Solicitud de ayuda en caso de presentar dificultades con algún hito
Start_date	Date	Fecha de inicio para un hito
Finish_date	Date	Fecha de culminación para un hito
IdArea	Int	Id del área a la cual pertenece cada hito Clave Foránea
idPriority	Int	Id de la prioridad que tiene cada Hito Clave Foránea
idHierarchy	Int	Id de la jerarquía que tiene cada Hito Clave Foránea
idStatus	Int	Id del estado en que se encuentra cada Hito Clave Foránea

Tabla 14. Definición de la tabla Hito.

Tabla: MemberHito

Esta tabla nace por la relación muchos a muchos que existe entre los miembros de una startup y los hitos.

Nombre	Tipo	Descripción
Id	Int unsigned	Clave Primaria
idHito	Int unsigned	Id del hito Clave Foránea
idMember	Int unsigned	Id del miembro asignado al Hito Clave Foránea

Tabla 15. Definición de la tabla MemberHito.

Tabla: Arc

Tabla encargada de almacenar los datos de los arcos del grafo.

Nombre	Tipo	Descripción
Id	Int unsigned	Clave Primaria
idFinal	Int unsigned	Id del hito final
idStart	Int unsigned	Id del hito inicial

Tabla 16. Definición de la tabla Arc.

Tabla: Área

Almacena los datos correspondientes a las áreas a las cuales pueden pertenecer los hitos

Nombre	Tipo	Descripción
Id	Int unsigned	Clave Primaria
Name	Char	Nombre del área

Tabla 17. Definición de la tabla Área.

Tabla: Priority

Almacena los datos correspondientes a la prioridad que tiene cada hito

Nombre	Tipo	Descripción
Id	Int unsigned	Clave Primaria
Name	Char	Nombre de la prioridad

Tabla 18. Definición de la tabla Priority.

Tabla: Hierarchy

Almacena los datos correspondientes a la jerarquía que tiene cada hito

Nombre	Tipo	Descripción
Id	Int unsigned	Clave Primaria
Name	Char	Nombre de la jerarquía

Tabla 19. Definición de la tabla Hierarchy.

Tabla: Status

Almacena los datos correspondientes a la jerarquía que tiene cada hito

Nombre	Tipo	Descripción
Id	Int unsigned	Clave Primaria
Name	Char	Nombre del estado en que se encuentra cada Hito

Tabla 20. Definición de la tabla Status.

Índice

1. Primer Capítulo	1
1.1 Planteamiento del problema.....	1
1.2 Justificación	3
1.3 Objetivo General.....	3
1.4 Objetivos Específicos.....	3
1.5 Alcance.....	4
1.6 Antecedentes.....	4
1.7 Metodología.....	5
1.8 Cronograma de Actividades.....	5
1.9 Cronograma de Evaluaciones.....	5
2. Capítulo 2.....	7
2.1 ¿Qué es una Startup?.....	7
2.2 ¿Cómo está formada una Startup?.....	7
2.3 ¿Qué busca una Startup?.....	7
2.4 ¿Qué es un MVP?.....	8
2.5 ¿Cuán exitosas son las Startups?.....	8
2.6 ¿Qué es un gestor de proyectos?.....	8
2.7 ¿Qué es un grafo dirigido?.....	8
2.8 ¿Qué es una línea de tiempo?.....	9
2.9 ¿Qué es una herramienta web?.....	9
2.10 ¿Qué es una aceleradora de startups?.....	9
3. Capítulo 3.....	10
3.1 Diseño del sistema.....	10
3.2 Requisitos del sistema.....	13
3.2.1 Requisitos funcionales.....	13
3.2.2 Requisitos no funcionales.....	14
3.3 Modelo del sistema.....	14
3.3.1 Modelo de actores.....	14
3.3.2 Modelado de asignaciones.....	15
3.3.3 Modelado de áreas de trabajo.....	16
3.3.4 Interacción.....	16
3.4 Casos de Uso.....	17
3.4.1 Gestionar MVP.....	18
3.4.2 Gestionar Área de Trabajo.....	18
3.4.3 Gestionar Hitos.....	19

3.4.4	Consultar MVP.....	20
3.4.5	Editar Hito.....	21
3.4.6	Eliminar Hito.....	22
3.4.7	Actualizar línea de tiempo.....	22
3.4.8	Conectar Hitos.....	23
3.4.9	Actualizar Dashboard.....	23
3.4.10	Crear Hito.....	24
4.	Capítulo 4.....	25
4.1	Justificación del diseño del sistema.....	25
4.2	Diagrama de despliegue.....	26
4.3	Arquitectura de software.....	27
4.4	Modelo relacional de base de datos.....	28
4.5	Diagrama de clases.....	29
4.6	Diagrama de Paquetes.....	30
5.	Capítulo 5.....	31
5.1	Descripción de la Metodología.....	31
5.2	Desarrollo de GeMVP.....	31
5.3	Pruebas del sistema.....	33
6.	Capítulo 6.....	35
6.1	Manual de Usuario.....	35
7.	Capítulo 7.....	46
7.1	Conclusiones.....	46
7.2	Recomendaciones.....	47
	Bibliografía.....	48
	Apéndices.....	49
	Definiciones técnicas.....	49
	HTML.....	49
	Navegador Web.....	49
	¿Qué es un Framework?.....	49
	¿Qué es un lenguaje de programación?.....	50
	¿Qué es una base de datos?.....	50

	58
PHP.....	50
MVC.....	50
JavaScript.....	51
SQL.....	51
MySQL.....	51
Laragon.....	51
Laravel.....	51
vis.js.....	51
Código fuente de la aplicación.....	52
Registro de cambios.....	52
Estructura de la base de datos.....	52
Tabla Startup.....	52
Tabla MVP.....	52
Tabla Member.....	53
Tabla Hito.....	53
Tabla MemberHito.....	54
Tabla Arc.....	54
Tabla Area.....	54
Tabla Priority.....	55
Tabla Hierarchy.....	55
Tabla Status.....	55

Índice de figuras

1. Figura 1 GeMVP.....	11
2. Figura 2 Añadir nodo.....	11
3. Figura 3 Editar nodo.....	12
4. Figura 4 Eliminar nodo.....	12
5. Figura 5 desarrollo del MVP.....	13
6. Figura 6 Modelo de actores.....	15
7. Figura 7 Modelo de casos de uso.....	17
8. Figura 8 Modelo de Casos de Uso. Gestionar Hitos.....	21
9. Figura 9 Diagrama de despliegue.....	26
10. Figura 10 Arquitectura usada por Laravel.....	27
11. Figura 11 Diagrama de clases.....	29
12. Figura 29. Diagrama de paquetes.....	30
13. Figura 12 Pantalla Inicial.....	35
14. Figura 13 Botón de agregar hito.....	36
15. Figura 14 Dashboard.....	36
16. Figura 15 Nodo Nuevo.....	37
17. Figura 16 Nodo nuevo detallado.....	38
18. Figura 17 Visualización del nombre y descripción de cada nodo.....	38
19. Figura 18 Ventana emergente para gestionar nodos.....	39
20. Figura 19 Editar nodo.....	40
21. Figura 20 Botón de guardar.....	40
22. Figura 21 Nodo con su información correspondiente.....	41
23. Figura 22 Línea de tiempo actualizada.....	41
24. Figura 23 Casilla "Previo".....	42
25. Figura 24 Grafo de actividades.....	42
26. Figura 25 Línea de tiempo actualizada con dos hitos.....	43
27. Figura 26 Flechas virtuales para visualización de grafo.....	43
28. Figura 27 Herramienta de visualización botones de zoom.....	44
29. Figura 28 Vista general.....	45

Índice de tablas

1. Tabla 1. Casos de Uso: Gestionar MVP.....	18
2. Tabla 2 Casos de Uso: Gestionar Área de Trabajo.....	19
3. Tabla 3 Casos de Uso: Gestionar Hitos.....	19
4. Tabla 4 Casos de Uso: Consultar MVP.....	20
5. Tabla 5 Casos de Uso: Editar Hito.....	22
6. Tabla 6 Casos de Uso: Eliminar Hito.....	22
7. Tabla 7 Casos de Uso: Actualizar línea de tiempo.....	23
8. Tabla 8 Casos de Uso: Conectar Hitos.....	23
9. Tabla 9 Casos de Uso: Actualizar Dashboard.....	24
10. Tabla 10 Casos de Uso: Crear Hito.....	24
11. Tabla 11 Definición de la tabla StartUp.....	52
12. Tabla 12 Definición de la tabla MVP.....	52
13. Tabla 13 Definición de la tabla Member.....	53
14. Tabla 14 Definición de la tabla Hito.....	53
15. Tabla 15 Definición de la tabla MemberHito.....	54
16. Tabla 16 Definición de la tabla Arc.....	54
17. Tabla 17 Definición de la tabla Area.....	54
18. Tabla 18 Definición de la tabla Priority.....	55
19. Tabla 19 Definición de la tabla Hierarchy.....	55
20. Tabla 20 Definición de la tabla Status.....	55