

# Propuesta del Proyecto

Andrea Velásquez Ávila <sup>[415033]</sup>, Cristhian Yovanny Monsalve Camacho <sup>[400739]</sup> and  
Antonio Jose Donis Hung <sup>[408397]</sup>

Universidad Pontificia Bolivariana, Bucaramanga Santander, Colombia

## 1 Información General

- **Título:** Laboratorio Virtual para el aprendizaje de Estructuras de Datos (LVaED)
- **Integrantes:** Andrea Velásquez Ávila, Cristhian Yovanny Monsalve Camacho, Antonio Jose Donis Hung
- **Institución:** Universidad Pontificia Bolivariana Seccional Bucaramanga.
- **Escuela:** Ingeniería.
- **Facultad:** Ingeniería de Sistemas e Informática. Bucaramanga. 2021.

## 2 Especificación de la situación problemática

Planteado por el profesor e Ingeniero Lenin Javier Serrano Gil:

“La Universidad Pontificia Bolivariana seccional Bucaramanga en su estrategia de continuidad para hacer frente al problema del covid19, ha centrado todos sus esfuerzos en el desarrollo de herramientas para la enseñanza virtual. La UPB solicita una herramienta web que promueva los contenidos del curso de estructuras de datos. Dicha herramienta llevara por título Laboratorio Virtual para el aprendizaje de Estructuras de Datos (LVaED) y es descrita a continuación. LVaED es una herramienta web en la que los usuarios pueden conocer diferentes estructuras de datos. Para esto LVaED tendrá tres módulos.

**Módulo de presentación**, en este se muestra de forma atractiva la teoría de cada estructura de datos con dos ejemplos prácticos y un caso de estudio. La presentación debe exponer la introducción, el concepto que la define, las operaciones que se permiten, las ventajas y los inconvenientes.

**Módulo de ejemplos**, aquí el sistema presenta un ejemplo de código debidamente documentado de las estructuras en los lenguajes Java, C ++ y Python. Donde explique el paradigma y las diferencias de codificación.

**Módulo de transformaciones**, LVaED, permitiría al usuario escribir o adjuntar el código fuente de una estructura de datos para hacer una representación sobre una instancia en un modelo. La base del modelo son los grafos y debe estar previamente definido en un metalenguaje. LVaED también permitiría la operación inversa. Es decir, el usuario podrá crear la instancia de la estructura de datos en un modelo y el sistema podrá generar al menos la interfaz en código para ella.

Las estructuras de datos mínimas que se podrán consultar son: Listas encadenadas sencillas, listas doblemente encadenadas, listas circulares encadenadas, pilas, colas, colas de prioridad, arboles binarios, árbol binario de búsqueda, arboles AVL, arboles B”

En ese contexto, nace la siguiente pregunta de investigación.

¿Cómo por medio de la ingeniería de software y de sistemas e informática, se puede desarrollar una herramienta web que cumpla con todos o al menos los requerimientos mínimos especificados por el Ingeniero Lenin Javier Serrano Gil?

### 3 Marco conceptual

- **Python:** Python es un lenguaje multiparadigma (funcional, orientado a objetos, estructurado...) de tipado dinámico, con recolector de basura. Diseñado por Guido van Rossum. [1] [2]
- **Java:** Java es un lenguaje de programación orientado a objetos, de fuerte tipado, con recolector de basura. [3] [4]
- **C++:** C++ es un lenguaje multiparadigma (funcional, estructurado, orientado a objetos...) de fuerte tipado. [5]
- **Flask:** Flask es un Framework para el desarrollo de sitios web implementado en Python para Python. Entre funcionalidades está el Path routing, templates y muchas otras que ayudan a los -desarrolladores con la implementación de servicios web. [6]
- **HTML:** HTML es un markup language diseñado para documentos mostrados en la web. [7]
- **CSS:** CSS es un Style sheet language diseñado para modificar la presentación de los documentos escritos en HTML. [8]
- **JavaScript:** JavaScript es un lenguaje de programación embeddable. Además, junto a HTML y CSS conforma una de las bases de la web contemporánea. [9]
- **Tipo de dato:** En computación, los tipos de datos se utilizan para representar secuencias de bits de diferentes formas, también se utilizan como herramientas de desarrollo debido a que implementan nuevos mecanismos para relacionar información. [10]
- **Frontend:** El Frontend tiene muchos significados, usualmente se le asocia en computación, con cualquier pieza de código que es visible y sirve de interfaz para el usuario, ejemplo de esto son los sitios web, la interfaz de un programa de escritorio, las interfaces de línea de comandos, etc. [11]
- **Backend:** El Backend tiene muchos significados, usualmente se le asocia en computación, con cualquier pieza de código que es invisible al usuario, usualmente, es la caja negra de los programas, ya que alguna interfaz de usuario interactúa con ella. Ejemplo de esta son, los REST API, librerías, RPC, etc.
- **REST API:** un REST API es una interfaz de programación de aplicaciones basada en el concepto de REST, estas interfases suelen recibir pedidos utilizando el protocolo HTTP y suelen retornar los resultados con el mismo y en alguno de los siguientes formatos: JSON, XML o puro texto plano. [12]
- **Abstract syntax tree:** Es un lenguaje intermedio en el cual se debería poder representar cualquier lenguaje de programación. Es una de las herramientas que utilizan

los compiladores para trabajar código y fue diseñado con el fin de poder estructurarlo. [13] [14]

- **Kanban:** Kanban es un modelo, metodología y estrategia para el desarrollo rápido de software, se basa principalmente en la implementación de funcionalidades a modo de tareas y fases, la cuales suelen ser por mínimo, Por hacer, en desarrollo, en prueba y terminadas. [15] [16]

## 4 Objetivos

### 4.1 Objetivo general

Utilizando los conceptos, metodologías y herramientas estudiadas en clase, diseñar, desarrollar y poner en producción una herramienta web que cumpla con todos los requisitos o al menos los mínimos especificados por el Ingeniero Lenin Javier Serrano Gil

### 4.2 Objetivos específicos

- Recopilar los requisitos del software especificados por el Ingeniero Lenin Javier Serrano Gil.
- Diseñar el sistema que conformara la herramienta web.
- Diseñar la interfaz de usuario que usara la herramienta web
- Implementar en el Backend de la herramienta web
- Implementar el Frontend de la herramienta web
- Desplegar en una máquina virtual local la herramienta web, de ser posible en un servidor.

## 5 Justificación

Con la gran demanda de páginas web y microservicios web es de suma importancia tanto para los ingenieros como los hobbyistas, entender por cuales procesos de diseño, desarrollo y despliegue, servicios web de alta demanda suelen pasar. Por otro lado, teniendo en cuenta que es una necesidad de la universidad, también se busca que la Universidad Pontificia Bolivariana de Bucaramanga, tenga todas las herramientas necesarias que le permitan a sus estudiantes de Ingeniería de sistemas, tener éxito en sus estudios, debido a que la herramienta web resultante está planteada principalmente para ello.

## 6 Metodología

Nosotros utilizaremos una variante propia del modelo Kanban, variante porque no es exactamente Kanban. Principalmente porque de cada funcionalidad implementada, si dicha implementación resulto compleja, arbitrariamente se le hará una cantidad de

iteraciones con el fin de confirmar que su funcionalidad es correcta. La planeación general de la implementación del proyecto es la siguiente:

- **Diseño:** Esto conforma al diseño general del proyecto. Aquí se planeará la vista general de la aplicación.
- **Desarrollo:** La propia implementación de las funcionalidades. Todo lo que conlleva a desarrollo de software y programación.
- **Validación:** La etapa de pruebas por las que pasara la aplicación en general.
- **Despliegue:** Finalmente el despliegue en una máquina virtual.

Cada vez que se vaya a implementar un componente o una funcionalidad, se deberá pasar por cada una de estas etapas:

- **Diseño:** Cada componente y funcionalidad de la aplicación será diseñada de modo que satisfaga todas las necesidades correspondientes.
- **Implementación:** Esta es la etapa de codificación.
- **Validación:** La etapa de validación del componente o funcionalidad implementada.

## 7 Cronograma de Actividades

### 7.1 Primer corte

Semana	Compromisos	Responsables	Porcentaje
1	Definición y presentación de los proyectos integra- dores o de aula	docentes	
	Jornadas metodo- lógicas para elabo- ración de propues- tas (actividad que se realizara con el profesor lí- der)	docentes y estu- diantes	
2	Entrega de pro- puestas de desa- rrollo de los pro- yectos	estudiantes	Propuesta: 30%
3	Jornadas de califi- cación, realimen- tación de las pro- puestas y de apoyo para el avance de los proyectos	docentes y estu- diantes	

5	Entrega de primer informe de proyectos	estudiantes	Informe: 20%
6	Primer seguimiento de resultados, realimentación del informe y de apoyo para el avance de proyectos	estudiantes	
7	Entrega de segundo informe de proyectos	docentes y estudiantes	Informe: 20%
9	Segundo seguimiento de resultados, realimentación del informe y de apoyo para el avance de proyectos	docentes y estudiantes	

## 7.2 Segundo corte

Semana	Compromisos	Responsables	Porcentaje
11	Entrega de tercer informe de proyectos	estudiantes	20%
13	Pre-sustentaciones y entrega final del proyecto (resultados, informe, software y demás entregables)	docentes y estudiantes	Pre-sustentación_10% Informe Final: 25% Producto: 25%
14	Jornada de socialización de proyectos integradores y de aula	docentes y estudiantes	sustentación: 20%
16	Entrega de notas a estudiantes	Docentes	

## Referencias

1. P. community, «Python,» python.org, [En línea]. Available: <https://www.python.org/>. [Último acceso: 27 1 2021].
2. W. community, «Python (programming language),» wikipedia.org, [En línea]. Available: [https://en.wikipedia.org/wiki/Python\\_\(programming\\_language\)](https://en.wikipedia.org/wiki/Python_(programming_language)). [Último acceso: 27 1 2021].
3. Oracle, «Java,» Oracle, [En línea]. Available: <https://www.java.com/en/>. [Último acceso: 27 1 2021].
4. W. community, «Java (programming language),» wikipedia.org, [En línea]. Available: [https://en.wikipedia.org/wiki/Java\\_\(programming\\_language\)](https://en.wikipedia.org/wiki/Java_(programming_language)). [Último acceso: 27 1 2021].
5. W. community, «C++,» wikipedia.org, [En línea]. Available: <https://en.wikipedia.org/wiki/C%2B%2B>. [Último acceso: 27 1 2021].
6. F. maintainers, «Flask,» flask.palletsprojects.com, [En línea]. Available: <https://flask.palletsprojects.com/en/1.1.x/>. [Último acceso: 27 1 2021].
7. W. community, «HTML,» wikipedia.org, [En línea]. Available: <https://en.wikipedia.org/wiki/HTML>. [Último acceso: 27 1 2021].
8. W. community, «CSS,» wikipedia.org, [En línea]. Available: <https://en.wikipedia.org/wiki/CSS>. [Último acceso: 27 1 2021].
9. W. community, «JavaScript,» wikipedia.org, [En línea]. Available: <https://en.wikipedia.org/wiki/JavaScript>. [Último acceso: 27 1 2021].
10. W. community, «Data type,» wikipedia.org, [En línea]. Available: [https://en.wikipedia.org/wiki/Data\\_type](https://en.wikipedia.org/wiki/Data_type). [Último acceso: 27 1 2021].
11. W. community, «Front-end web development,» wikipedia.org, [En línea]. Available: [https://en.wikipedia.org/wiki/Front-end\\_web\\_development](https://en.wikipedia.org/wiki/Front-end_web_development). [Último acceso: 27 1 2021].
12. R. team, «What is a REST API?,» RedHat, [En línea]. Available: <https://www.redhat.com/en/topics/api/what-is-a-rest-api>. [Último acceso: 27 1 2021].
13. W. community, «Abstract syntax tree,» wikipedia.org, [En línea]. Available: [https://en.wikipedia.org/wiki/Abstract\\_syntax\\_tree](https://en.wikipedia.org/wiki/Abstract_syntax_tree). [Último acceso: 27 1 2021].
14. D. KUNDEL, «ASTs - What are they and how to use them,» twilio, [En línea]. Available: <https://www.twilio.com/blog/abstract-syntax-trees>. [Último acceso: 27 1 2021].
15. J. team, «Kanban Tools,» jetBrains, [En línea]. Available: <https://www.jetbrains.com/youtrack/agile/kanban/>. [Último acceso: 27 1 2021].
16. W. community, «Kaban (development),» wikipedia.org, [En línea]. Available: [https://en.wikipedia.org/wiki/Kanban\\_\(development\)](https://en.wikipedia.org/wiki/Kanban_(development)). [Último acceso: 27 1 2021].