

**Desarrollo de un sistema educativo multimedia orientado a la web para optimizar
aprendizajes en cultivos de tejidos vegetales**



**NÉSTOR EDUARDO SUAT ROJAS
DAVID RICARDO GRANDAS CÁRDENAS**

**UNIVERSIDAD DE LOS LLANOS
FACULTAD DE CIENCIAS BÁSICAS E INGENIERÍA
INGENIERÍA DE SISTEMAS
VILLAVICENCIO
2016**

MANUAL DEL TÉCNICO

NÉSTOR EDUARDO SUAT ROJAS
DAVID RICARDO GRANDAS CÁRDENAS

UNIVERSIDAD DE LOS LLANOS
FACULTAD DE CIENCIAS BÁSICAS E INGENIERÍA
INGENIERÍA DE SISTEMAS
VILLAVICENCIO
2016

CONTENIDO

	Pág.
1. Presentación	5
2. Descripción del sistema	6
2.1 Características del servidor	6
2.2 Tecnologías usadas	7
2.3 Dependencias o Package.json	9
3. Requerimientos	10
3.1 Usuarios involucrados	10
3.2 Funcionales	11
3.3 No funcionales	14
4. Diagrama de navegación	15
5. Diagramas de casos de uso	15
5.1 Selección de capítulo	16
Casos de Uso	16
Fichas de Caso de uso	16
5.2 Selección de actividades	17
Casos de uso	17
Fichas de Caso de uso	17
6. Estructura de archivos	17
6.1 Árbol de directorio	18
6.2 Estructura archivos JSON	18
7. Arrancar Servidor	19

1. PRESENTACIÓN

La demanda de profesionales en el área de Biotecnología o **Cultivo de Tejidos Vegetales (C.T.V)** en Colombia por empresas e instituciones para el desarrollo de investigaciones y proyectos que permiten el estudio de biología molecular, química, bioquímica, fisiología, genética y la obtención de productos que potencia la agricultura y la conservación de especies cultivables como silvestres es una realidad creciente en las últimas décadas (Orozco L, 2006; Gonzalez *et al.*, 2010).

Por tal motivo, la formación y el perfil de los estudiantes de ingeniería agronomía (I.A.) y Lic. en Producción Agropecuaria (L.P.A) debe estar enfocado a ocupar estos nuevos espacios; actualmente la ola mundial de la economía esta basada en el conocimiento de este tipo de productos (Hernández, 2008). No obstante, existen falencias en el aprendizaje por parte de los estudiantes en cuanto a las actividades de laboratorio de cultivo de tejidos vegetales, desde la perspectiva teórica-práctica (Amórtegui *et al.*, 2010). Entonces, ¿Qué estrategia metacognitiva, regulada por un modelo pedagógico, permitirá que los educandos de L.P.A e I.A, utilizando el laboratorio virtual de cultivo de tejidos, comprendan mejor los saberes de manejo de los componentes teórico-prácticos en el área de Biotecnología?

Las Tecnologías de la información y Comunicación (TIC) fomenta la cultura informática en busca de generar ambientes virtuales de aprendizaje que se ofrezcan como estrategia para fortalecer la enseñanza (Guzmán, 2006), en particular en CTV; es un área por abordar por el interés que presenta el país. En este perfil de incentivación de manejo de TIC, a través de software aplicativos se han sumado Instituciones Educativas y Universidades de Norteamérica (Carnoy, 2004), Centro américa (Romero *et al.*, 2010) y Sur américa, ejemplo, Colombia con el Grupo de investigación de Educación de Ambientes Virtuales que integra material multimedia (Informe, 2010) obteniendo resultados satisfactorios de aprendizaje por parte de los usuarios. Empero, el sistema educativo casi no utiliza las TIC para gestionar la calidad de sus resultados y aumentar la productividad de los profesores (Carnoy, 2004).

Consecuente con lo anterior, se proyectó el diseño, desarrollo e implementación de un software educativo multimedia de C. T. V con énfasis para promover el aprendizaje en componentes de biotecnología en los estudiantes de Ingeniería Agronómica y Lic. en Producción Agropecuaria y de apoyo para los docentes.

Sin embargo, este tipo de estrategias se refuerzan con la integración de modelos pedagógicos pertinentes al diseño de material didáctico multimedia. La integración de la estrategia metacognitiva, regulada por un modelo pedagógica que la potencie, involucra un trabajo de parte del aprendiz, por lo que se busca alcance una autorregulación del proceso cognitivo (Osses & Jaramillo, 2008). Sin dejar a un lado al orientador del proceso que es el docente experto. En cierto modo se busca aunar la metodología de diseño de software con la metacognitiva, y a su vez esta última en la aplicación que le van a dar los estudiantes y el docente en su rol respectivo, con el fin de que cada usuario (Educando) del software, incremente su aprendizaje.

2. DESCRIPCIÓN DEL SISTEMA

El sistema esta elaborado con las ultimas tecnologías en programa de servidores para el desarrollo de aplicaciones en tiempo real y escalables para la elaboración de aplicaciones isomorficas, su diseño de interfaz fue orientado basándose en las pautas y normas para mejorar la experiencia de usuario a través de la navegación de sus funciones.

En este software orientado a la web se ha utilizado lo ultimo de librerías para la renderización de las paginas en respuesta de servidor, tiene una forma especial de hacerlo con el fin de optimizar la carga del servidor y cargar algunas funciones en el lado del cliente, al igual que servicio que marcan la pauta actualmente como Facebook, Instragram, Twitter y Gmail, entre los mas reconocidos.

El sistema maneja un solo rol, el cual se basa al acceso a la información y/o material didáctico en el aprendizaje de cultivos de tejidos vegetales; su contenido esta dividido por 7 capítulos, los cuales le llamamos durante su desarrollo como Módulos, que se diferencian por su contenido sobre la temática.

2.1. Características del servidor

Los directorios del proyecto se podrán alojar en cualquier servidor linux o windows que tenga instalado la versión de NodeJS v0.12.4 y babe-node v5.6.14, este ultimo teniendo en cuenta que se esta usando el ultimo estándar de Javascript ES6 o EcmaScript2015.

Para el funcionamiento del servidor solo se necesita el archivo app.js que se encuentra dentro de la carpeta /public, gracias a la herramienta Browserify todo el código fuente es generado en este único archivo, inclusive las librerías que se utilizan, optimizando así la respuesta del servidor, a continuación las carpetas y archivos que se necesitan cargar en servidor para su correcto funcionamiento.

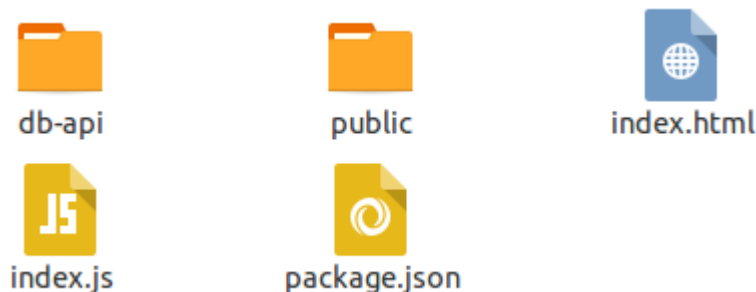


Ilustración 1: Archivos y carpetas necesarias para

2.2 Tecnologías usadas

HTML5

Es la última versión de la norma que define HTML, con nuevos atributos y comportamientos para enriquecer la experiencia y el desarrollo de los sitios Web.

Javascript

Al igual que HTML, Javascript es un lenguaje de programación que se puede utilizar para construir sitios Web y para hacerlos más interactivos. Usualmente se habla de este lenguaje para programar en el lado del cliente, en este proyecto también se uso este lenguaje en el lado del servidor.

CSS3

Hojas de Estilo en Cascada (Cascading Style Sheets), es un mecanismo simple que describe cómo se va a mostrar un documento en la pantalla, o cómo se va a imprimir, o incluso cómo va a ser pronunciada la información presente en ese documento a través de un dispositivo de lectura. Esta forma de descripción de estilos ofrece a los desarrolladores el control total sobre estilo y formato de sus documentos.

JQUERY

Es un una biblioteca o framework de Javascript que permite simplificar la manera de interactuar con los documentos HTML, permitiendo manejar eventos, desarrollar animaciones y agregar interacción con la tecnología AJAX a páginas web. jQuery, al igual que otras librerías, ofrece una serie de funcionalidades basadas en Javascript que de otra manera requerirían de mucho más código.

JSX

Es un pseudolenguaje que facilita el desarrollo de aplicaciones web con su sintaxis para crear elementos en el DOM.

Recordemos que ReactJS se enfoca en la visualización y literalmente debemos armar el HTML que deseamos directamente en **JSX**, con el fin de describir nuestros componentes por medio de etiquetas y de una sintaxis muy parecida a la de HTML, que es compilada a Javascript.

ReactJS

Es una biblioteca escrita en JavaScript, desarrollada en Facebook para facilitar la creación de componentes interactivos, reutilizables, para interfaces de usuario. Se utiliza en Facebook para la producción de componentes, e Instagram está escrito enteramente en React. Uno de sus puntos más destacados, es que no sólo se utiliza en el lado del cliente, sino que también se puede representar en el servidor, y trabajar juntos.

React Router

React Router es una librería para gestionar rutas en aplicaciones que utilicen ReactJS. Está inspirada en el sistema de enrutado de Ember.js y su forma de manejar las rutas es un poco diferente de lo que podemos ver en otros sistemas, como ASP.NET MVC, AngularJS, Express o Compojure.

Bootstrap

Bootstrap, es un framework originalmente creado por Twitter, que permite crear interfaces web con CSS y JavaScript, cuya particularidad es la de adaptar la interfaz del sitio web al tamaño del dispositivo en que se visualice. Es decir, el sitio web se adapta automáticamente al tamaño de una PC, una Tablet u otro dispositivo. Esta técnica de diseño y desarrollo se conoce como “**responsive design**” o diseño adaptativo.

NodeJS

Node es un intérprete Javascript del lado del servidor que cambia la noción de cómo debería trabajar un servidor. Su meta es permitir a un programador construir aplicaciones altamente escalables y escribir código que maneje decenas de miles de conexiones simultáneas en una sólo una máquina física.

Babel y babel-node

Al usar el estándar ES6 los navegadores actuales no tienen soporte total de sus funcionalidades incorporadas para ello se usa un transpilador como Babel para que traduzca el código al estándar de javascript anterior. Igualmente babel-node es para dar soporte en el servidor NodeJs para que ejecute el código sin ningún problema.

Browserify

Browserify es una librería de Node.js, escrita por substack, uno de los mayores contribuidores al core de Node.js y con mayor número de módulos publicados en NPM.

Nos permite escribir código JavaScript del cliente, como si estuviésemos programando en Node.js, es decir, como por ahora no tenemos módulos de forma nativa en JavaScript (hasta que se implante el estándar ECMAScript6 que si los trae) hay librerías que se encargan de

imitar ese comportamiento (Caso de Require.js hace un tiempo).

Con esta librería, podemos instalar módulos para el cliente, con NPM, al igual que hacíamos con bower, pero ahora, podemos llamarlos desde nuestro código con la palabra reservada `require`, al igual que en Node.js

Y también nos permite tener un sólo archivo .js en el cliente, ya de inicio, olvidándonos de tener que importar todas las librerías y scripts utilizados en el HTML con `<script src="..."></script>` y luego tener que concatenar etc...

También nos permite crear nuestros propios módulos con `module.exports` y llamarlos de nuevo desde nuestro código con `require`.

2.3 Dependencias o Package.json

En este archivo, que debe estar en la raíz del proyecto, va a quedar reflejada la configuración del proyecto de **Node** tales como: nombre del proyecto, autor, versión, dependencias, scripts y otros campos que se pueden detallar en la documentación del servicio de NPM.

Este archivo es necesario cuando usamos el servicio de NPM (Node Packaged Manager) para importar módulos y librerías desarrolladas por la comunidad de Node, en este archivo se encuentra nuestra configuración como se presenta a continuación:

```
{
  "name": "AppCTV",
  "version": "0.1.1",
  "description": "Laboratorio Virtual de Cultivo de Tejidos Vegetales - Universidad de los llanos",
  "autor": "Desarrollador: Néstor Suat - Documentador: David Grandas",
  "dependencies": {
    "react-bootstrap": "0.28.1",
    "babel": "5.8.34",
    "babelify": "^6.1.3",
    "browserify": "^11.0.1",
    "express": "^4.13.3",
    "jquery": "^2.1.4",
    "history": "1.13.1",
    "react": "0.14.3",
    "react-router": "1.0.2",
    "react-addons-css-transition-group": "0.14.3",
    "react-dom": "0.14.3",
    "react-router-bootstrap": "0.19.3",
    "uid": "^0.0.2"
  },
  "scripts": {
    "build": "browserify app.js > public/app.js -t babelify",
    "start": "babel-node index.js"
  }
}
```

Ilustración 2: configuración del archivo Package.json

3. REQUERIMIENTOS

3.1. Usuarios Involucrados

Usuario	Descripción
ESTUDIANTE	Estudiante del programa de Producción Agropecuaria e Ingeniería Agronómica

Tabla 1. Requerimiento Usuarios

3.2. Funcionales

ID del Requerimiento	RF_PASS_1
Nombre del Requerimiento	Listado de capítulos
Características	Acceso desde toda aplicación
Descripción del Requerimiento	El listado de los capítulos se encontrará presente en todas las vistas de la aplicación
Prioridad del Requerimiento	Media

Tabla 2. Requerimiento Funcional 1

ID del Requerimiento		RF_PASS_2	
Nombre del Requerimiento		Introducción de los capítulos	
Características		Acceso capítulo uno a siete	
Descripción del Requerimiento		Los capítulos uno a siete contara con una sección en la que se da una introducción a la temática que se verá posteriormente en el capítulo	
Prioridad del Requerimiento		Alta	

Tabla 3. Requerimiento Funcional 2

ID del Requerimiento		RF_PASS_3	
Nombre del Requerimiento		Contenido de los Capítulos	
Características		Se encontrara en los siete capítulos	
Descripción del Requerimiento		Los capítulos contarán con secciones en la que se visualizara el contenido de aprendizaje por el alumno	
Prioridad del Requerimiento		Alta	

Tabla 4. Requerimiento Funcional 3

ID del Requerimiento		RF_PASS_4	
Nombre del Requerimiento		Paginación de contenido	
Características		Relevancia del contenido	
Descripción del Requerimiento		La información deberá permitir la visualización por un método de paginación	
Prioridad del Requerimiento		Baja	

Tabla 5. Requerimiento Funcional 4

ID del Requerimiento		RF_PASS_5	
Nombre del Requerimiento		Animación	
Características		Animación de contenido	
Descripción del Requerimiento		En la transición entre los módulos debe contener una animación	
Prioridad del Requerimiento		Baja	

Tabla 6. Requerimiento Funcional 5

ID del Requerimiento		RF_PASS_6	
Nombre del Requerimiento		Galería	
Características		Acceso desde toda la aplicación	
Descripción del Requerimiento		Los capítulos contarán con secciones en la que se visualizara el contenido de los sub módulos aprendizaje por el alumno	
Prioridad del Requerimiento		Alta	

Tabla 7. Requerimiento Funcional 6

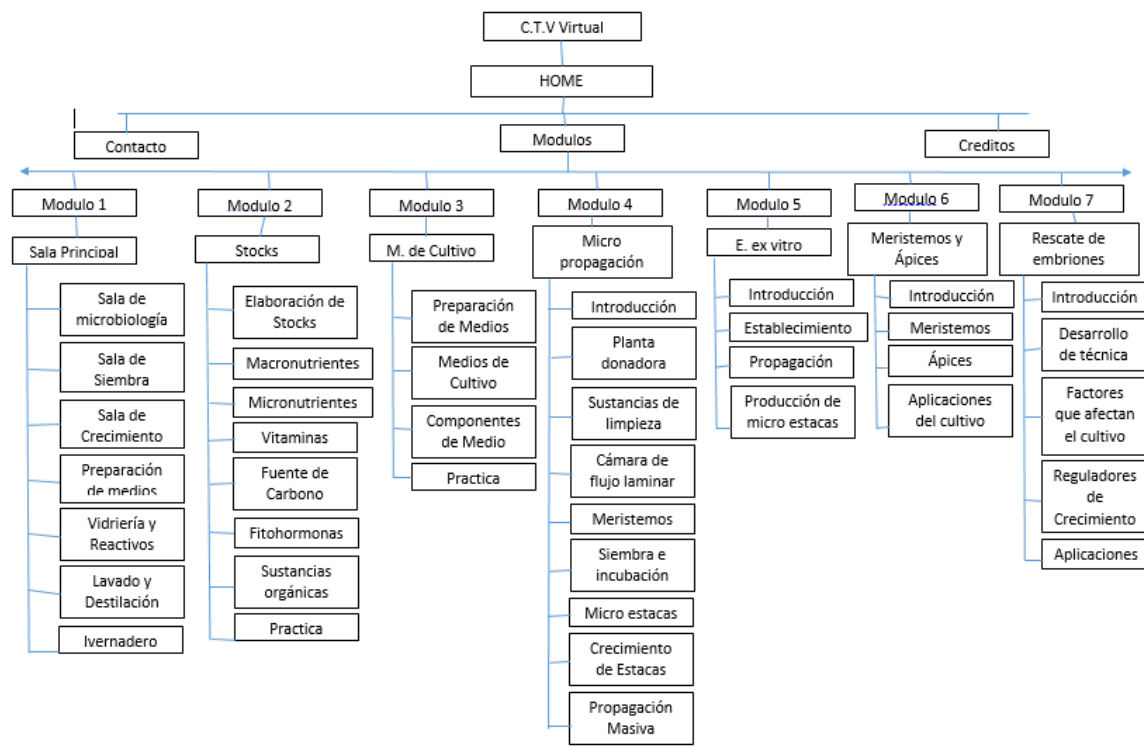
ID del Requerimiento		RF_PASS_7	
Nombre del Requerimiento		Colaboradores	
Características		Acceso desde toda la aplicación	
Descripción del Requerimiento		Existirá una sección en la que se pueda visualizar todos los integrantes del equipo de trabajo.	
Prioridad del Requerimiento		Alta	

Tabla 4. Requerimiento Funcional 7

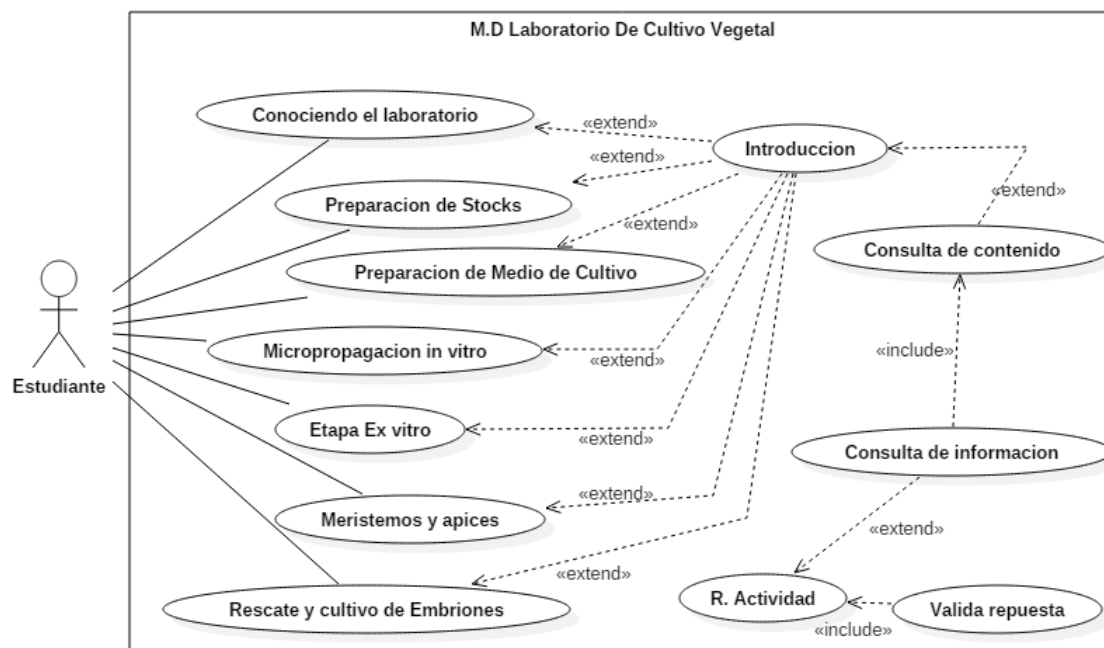
3.3. No Funcionales

- **RNF01.** El acceso a la aplicación se podrá realizar desde un navegador web que cumpla con los estándares provisto por el lenguaje html 5.
- **RNF02.** El aplicativo debe tener en cuenta los estandares de usabilidad para el desarrollo de un aplicativo de enseñanza.
- **RNF03.** La respuesta de la aplicación a la hora de mostrar el contenido no debe alargarse demasiado, obteniendo una respuesta rápida (se estima que con 10 segundos).
- **RNF04.** El sistema completo deberá ser posible almacenarlo en un disco. La aplicación no requerirá demasiado espacio (se estima que con 300 MB será suficiente).
- **RNF05.** El sistema debe ser fiable, puesto que un error de la aplicación puede costar la desmotivación por parte del alumno.
- **RNF06.** El usuario debe ser capaz de familiarizarse rápidamente con la interfaz gráfica del sistema, y para solicitar el servicio de domicilio lo hará en un tiempo promedio de 6 minutos de interacción con el sistema.

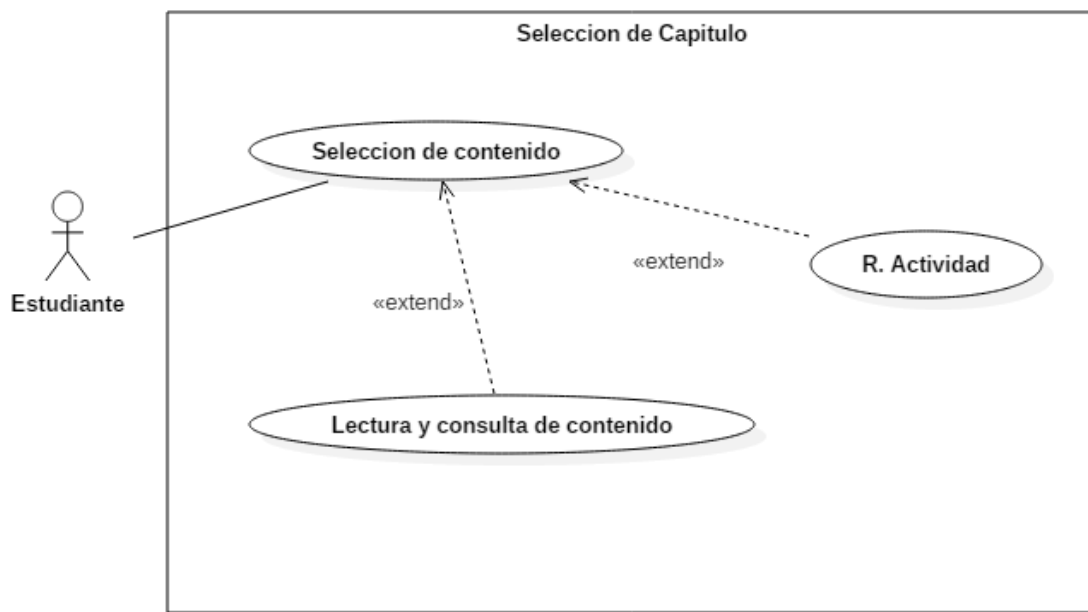
4. Diagrama de navegación



5. DIAGRAMA DE CASOS DE USO



UC1	Selección General
Versión	Final
Autor	David Ricardo Grandas
Descripción	Permite al usuario seleccionar el capítulo a estudiar entre los propuestos en el aplicativo.
Precondición	Introducción en la aplicación
Flujo Normal del Evento	
Paso	Acción
P1	El usuario en el aplicativo debe ubicar y seleccionar el capítulo a estudiar
P2	Selecciona dicho capítulo y lo visualiza
P3	Se direcciona al actor al capítulo seleccionado
Postcondición	Si se realizó correctamente el caso de uso, el actor accederá al capítulo seleccionado y continuara la navegación
Excepciones Flujo Alternativo de Eventos	
Paso	Acción
P1	El estudiante regresara a la selección de capítulo
P2	Nada
P3	Nada



UC2	Selección de Capitulo
Versión	Final
Autor	David Ricardo Grandas
Descripción	Permite al usuario seleccionar el subcapítulo a estudiar entre los propuestos en el aplicativo.
Precondición	Introducción en la aplicación
Flujo Normal del Evento	
Paso	Acción
P1	El usuario en el aplicativo debe ubicar y seleccionar el subcapítulo a estudiar
P2	Selecciona dicho capitulo y lo visualiza
P3	Se direcciona al actor al capítulo seleccionado
Postcondición	Si se realizó correctamente el caso de uso, el actor accederá al capítulo seleccionado y continuara la navegación
Excepciones Flujo Alternativo de Eventos	
Paso	Acción
P1	El estudiante regresara a la selección de capitulo
P2	Nada
P3	Nada

6. ESTRUCTURA DE ARCHIVOS

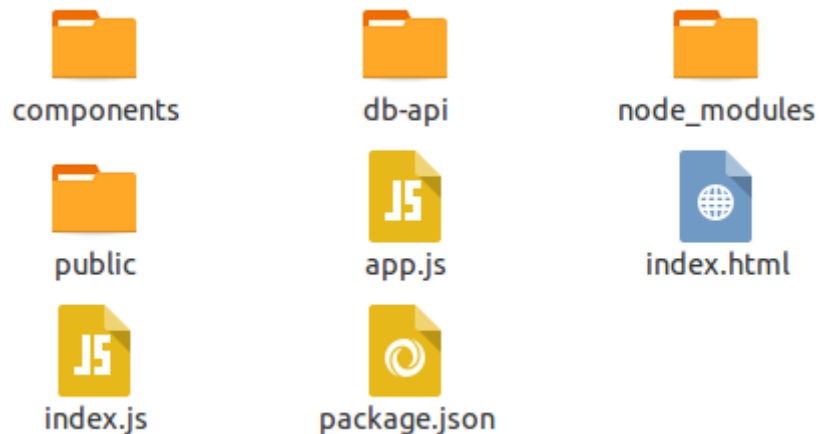


Ilustración 1: Estructura de archivos

6.1. Estructura de archivos.

components: Contiene todo el código fuente de la aplicación en JSX.

db-api: Contiene archivos json, con gran parte del contenido de la aplicación.

node_modules: Contiene las dependencias necesarias para el funcionamiento de la aplicación.

public: Contiene los archivos css, javascript (librerías) e imágenes necesarias, además está el código compilado por browserify.

app.js: archivo de rutas gestionado por la librería react-router.

index.html: archivo html de la aplicación.

index.js: archivo con la configuración para arrancar servidor NodeJs.

package.json: Archivo de configuración con información de la aplicación, juntos con la información de las dependencias necesarias.

6.2. Estructura archivo Json

Clave	Valor
Modulo	Valor numerico con el id del contenido para usarlo como key del componente al momento de renderizar.
Titulo	Valor alfanumerico con el titulo del contenido del componente
Descripción	Valor alfanumerico con el material que describe el contenido del componente.
Url	Valor alfanumerico para cargar como direccionaminto.

```
ContentModulos.js x
export default[
  {
    modulo: 1,
    titulo: 'Modulo 1. Conociendo el laboratorio',
    descripcion: 'Paseo Virtual. Encontramos secciones de entrada y salida, lavado y esterilización, cuarto de vidriería y reactivos.',
    url: 'modulo1/1'
  },
  {
    modulo: 2,
    titulo: 'Modulo 2. Preparación de Stocks',
    descripcion: 'Elaboración de stocks para usar como fuente de nutrimento para el medio de cultivo.',
    url: 'modulo/2'
  },
  {
    modulo: 3,
    titulo: 'Modulo 3. Preparación de Medios de Cultivo.',
    descripcion: 'Las plantas necesitan un soporte para su desarrollo radicular y sostenimiento.',
    url: 'modulo/3'
  },
  {
    modulo: 4,
    titulo: 'Modulo 4. MICROPROPAGACIÓN IN VITRO',
    descripcion: 'El cultivo de tejidos vegetales incluye diferente técnicas de micropropagación in vitro.',
    url: 'modulo/4'
  },
  {
    modulo: 5,
    titulo: 'Modulo 5. ETAPA EX VITRO',
    descripcion: 'Las vitroplantas poseen condiciones controladas y, con el tarro cerrado y sellado, su intercambio gaseoso es mínimo.',
    url: 'modulo/5'
  },
  {
    modulo: 6,
    titulo: 'Modulo 6. Preparación de Medios',
    descripcion: 'Elaboración de Medios.',
    url: 'modulo/6'
  },
  {
    modulo: 7,
    titulo: 'Modulo 7. Preparación de Medios',
    descripcion: 'Elaboración de Medios.',
    url: 'modulo/7'
  },
  {
    modulo: 8,

```

Ilustración 2: Estructura de código json

7. ARRANCAR SERVIDOR

Teniendo en cuenta que el servidor usado es NodeJs, dentro del archivo package.json se encuentran dos scripts para compilar el código fuente con Browserify y para arrancar el servidor node con transpilador babel.

A continuación son las dos instrucciones que se ejecutan con el gestor de paquetes NPM:

npm run build: ejecturar browserify, hacerlo cada vez que se modifique el código.

npm run start: Si no hay cambios, o si ya se modifico, este comando es para lanzar el servidor y se ejecutara la aplicación en el puerto 3000, la utilización de este puerto se puede configurar en el archivo index.js