

Camilo Andres Paredes Castellanos

Ficha 2848524 A

Servicio Nacional de Aprendizaje-SENA

Tecnologia de Analisis y Desarrollo de software

Informe Backend del proyecto

Elsa Maria Junca

Intructura

2025

Tabla de contenido

[Informe sobre el backend del módulo fabricación 4](#_ya9e3g80a8fm)

[Controlador 4](#_tg7tskre0uqq)

[ManufacturingController.php 4](#_96hflo6x5zo1)

[Modelo 7](#_qbl1igsbp0q7)

[Manufacturing.php: 7](#_4oxasvypqvee)

[Proceso de fabricación y registro de insumos 9](#_39pnfrdz7u3a)

[Inicialización de totales: 9](#_kp1xjv6vyp6n)

[Recorrido de ingredientes (foreach): 9](#_aagefo6bnvh1)

[Validación de stock: 9](#_7k91o9ofli84)

[Registro en récipe: 9](#_cz9n6raiu16w)

[Actualización de inventario: 9](#_gassh5smcge4)

[Cálculo y almacenamiento de totales: 9](#_w4cbr9p3nydv)

[Retorno del resultado: 10](#_bapfs2mrw64e)

[Recipe.php: 11](#_i4wt35befql3)

[Ilustración 1 Depencias ManufacturingControllers 4](#_m4wzsdv5aj51)

[Ilustración 2 Herencia Manufacturing Controllers 4](#_2gkqq0sexir1)

[Ilustración 3 Requerimientos Manufactruring controllers 5](#_s97x6a1u9nrp)

[Ilustración 4 Store Manufacturing controllers 6](#_xhlnvpna4o5q)

[Ilustración 5 Index Manufacturing Controllers 6](#_5s9x8p4s5cv)

[Ilustración 6 Destroy Manufacturing Controllers 7](#_wn0vpoxoqou9)

[Ilustración 7 Tabla Manufacturing Models 8](#_kot5eic4r0bj)

[Ilustración 8 Relaciones Manufacturing Models 8](#_r0hb23pxojej)

[Ilustración 9 CalculateLabour Manufacturing Models 9](#_gnp1f2iocwp0)

[Ilustración 10 addIngredients Manufacturing Models 10](#_5mpgpy8ui9bx)

[Ilustración 11 booted Recipe Models 11](#_83crlmy4igo2)

[Ilustración 12 getGramsAttribute Recipe Models 11](#_vhj1m7uynqut)

[Ilustración 13 getPriceReference Recipe models 12](#_6hmjxkh9puw)

[Ilustración 14 CalculatePriceSpent Recipe Models 12](#_a1mydolzdgw9)

[Ilustración 15 restoreStockInputs Recipe Models 13](#_55cmp2hj4ybq)

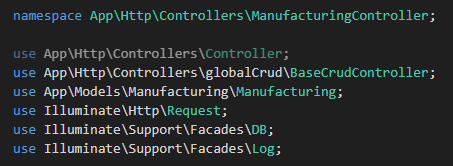
# Informe sobre el backend del módulo fabricación

# Controlador

## ManufacturingController.php

Se importan las independencias, que heredan las funcionalidades como lo es el BasecrudController es un CRUD global que en este caso voy a utilizar sus métodos en el controlador

Otras independencias sirven para acceder a los datos de petición HTTP por parte del request, otra sirve para hacer las transacciones a la base de datos y por último para registrar los errores en los logs



***Ilustración 1 Depencias ManufacturingControllers***

El ManufacturingController se extiende de la baseCrudController esto quiere decir que se hereda todo el método CRUD

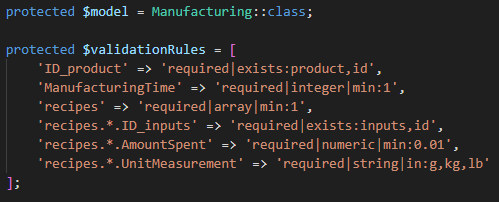


***Ilustración 2 Herencia Manufacturing Controllers***

Se indica que la propiedad modelo trabajara con el modelo manufacturing

Validaciones: esto indica que se debe de cumplir el request para que la fabricación se pueda crear, por ejemplo:

* ID\_product = es obligatorio de que tenga valores como el producto y su id, además tiene que existir
* ManufacturingTime = es requiere que números enteros y que sea mínimo uno
* recipes = tiene que ser un arreglo con al menos un elemento
* recipes.\*.ID\_inputs = requiere que los valores existan como los insumos y el id
* recipes.\*.AmountSpent = requiere un valor numérico y que sea mínimo a 0.01
* recipes.\*.UnitMeasurement = require un string Pero solo récipe unas letras definidas que son: g, kg, lb



***Ilustración 3 Requerimientos Manufactruring controllers***

Se crea una fabricación realizando el siguiente flujo:

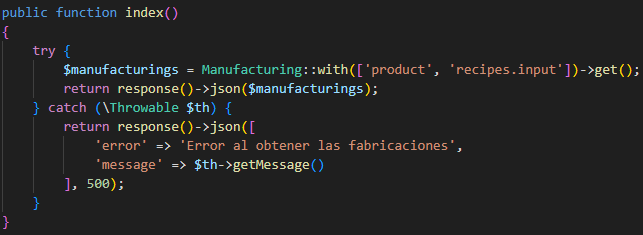
* se inicia una transacción
* se valida los datos del request
* Crea un registro en la tabla manufacturing
* Se calcula la mano de obra usando el método calculateLabour();
* Se agrega el insumo y se actualiza el stock
* Retorna un JSON con un mensaje de que la fabricación fue registrada
* Y si hay un error, hace un rollback mostrando que tipo de error es y lo registra en logs



***Ilustración 4 Store Manufacturing controllers***

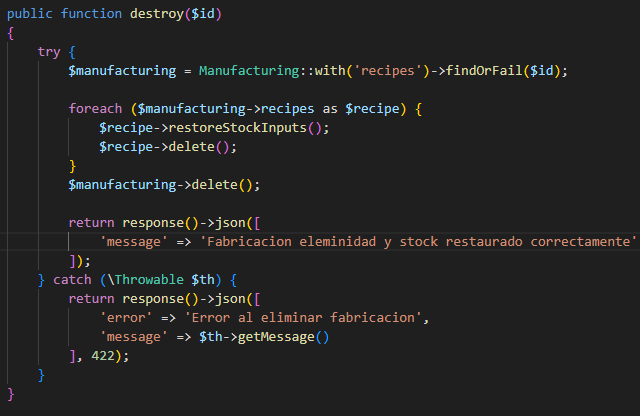
Se obtiene todas las fabricaciones, con los producto, recetas y insumos.

Si hay un error, devuelve un JSON con un mensaje y código 500



***Ilustración 5 Index Manufacturing Controllers***

Busca la fabricación por él ID, por ende, cada receta eliminada el stock se restaura y es eliminada de la base de datos y si todo se realizo correctamente devuelve un JSON confirmando lo realizado.



***Ilustración 6 Destroy Manufacturing Controllers***

# Modelo

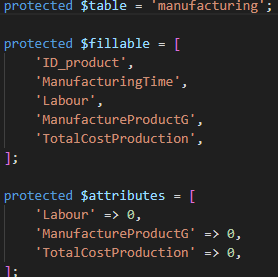
## Manufacturing.php:

Este modelo representa una fabricación en la base de datos, con la lógica para calcular la mano de obre, agrega ingredientes, verificar stock y actualizar costos

Indica los registros que se van a guardar en la tabla manufacturing

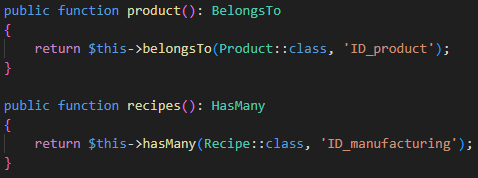
En la variable $fillable se indican los campos que se van a utilizar para crear o actualizar

En la variable $attirbutes son los valores que se dejan por defecto si no ahí registros llenos desde el comienzo



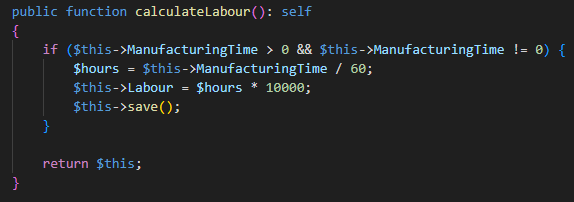
***Ilustración 7 Tabla Manufacturing Models***

Son relaciones en la cual se logra apreciar que la fabricación le pertenece a un solo producto y una fabricación tiene muchas recetas, en donde cada receta recetas representa uno o mas insumos utilizados



***Ilustración 8 Relaciones Manufacturing Models***

Convierte el tiempo de fabricación a horas y se condiciona de que el valor ingresado sea superior a 0. En la recolección de información se estableció que el costo de la mano de obra tiene una tarifa de 10.000 COP/hora y por ultimo se guarda el valor en el campo labour



***Ilustración 9 CalculateLabour Manufacturing Models***

Esta es el corazón de la lógica en insumos:

## Proceso de fabricación y registro de insumos

### Inicialización de totales:

* + Se establecen las variables de totales en cero antes de iniciar el procesamiento.

### Recorrido de ingredientes (foreach):

* + Se itera sobre cada ingrediente registrado.

### Validación de stock:

* Se verifica si la cantidad disponible supera los 200 gramos o si no existe suficiente inventario para el proceso.

### Registro en récipe:

* + Se guarda el insumo correspondiente en la tabla recipe.
  + Se utiliza el método del modelo Recipe para calcular el costo en base al **precio por unidad** y la **cantidad** utilizada.
  + El costo calculado se asigna al campo PriceQuantitySpent.

### Actualización de inventario:

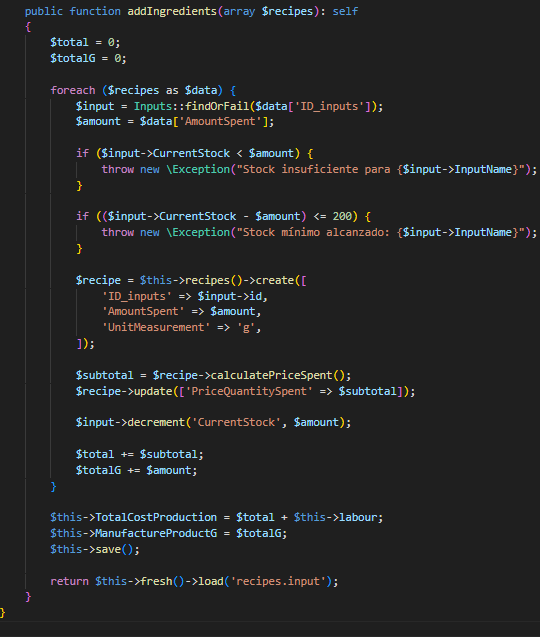
* + Se descuenta del stock la cantidad utilizada en el proceso.

### Cálculo y almacenamiento de totales:

* + Se suman los insumos y mano de obra a las variables acumuladoras.
  + Se registran los totales en la tabla de fabricación.

### Retorno del resultado:

* + Se devuelve el registro de fabricación actualizado, incluyendo las relaciones correspondientes.

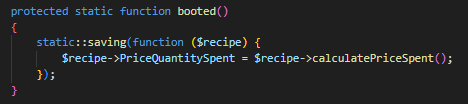


***Ilustración 10 addIngredients Manufacturing Models***

# Recipe.php:

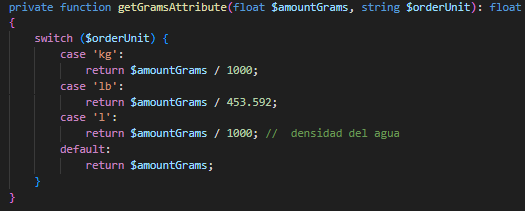
Récipe es el encargado de guardar el insumo utilizado, la fabricación, la cantidad utilizada, unidad de medida, costo de esa cantidad usada

Antes de guardar, se recalcula el costo usando calculatePriceSpent()



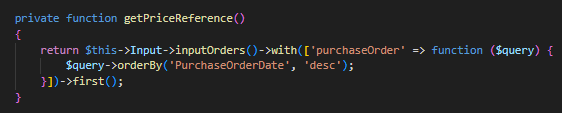
***Ilustración 11 booted Recipe Models***

Convierte la cantidad usada a la unidad correcta según como se compro el insumo



***Ilustración 12 getGramsAttribute Recipe Models***

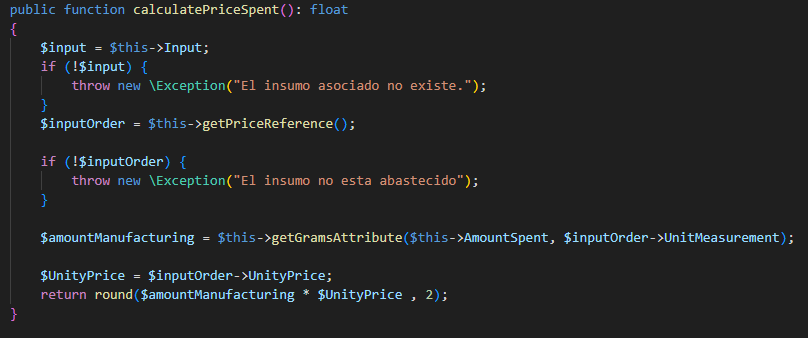
ordena por fechas de orden de compra de mas reciente a mas antigua, devolviendo el último precio



***Ilustración 13 getPriceReference Recipe models***

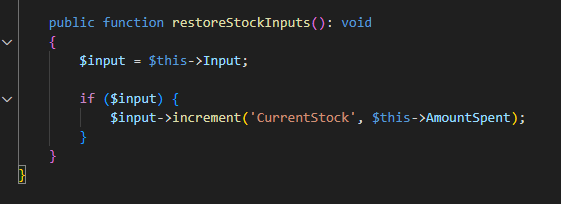
Verifica que si exista el insumo y tenga el precio registrado

Convierte la cantidad usada a la unidad de compra, después multiplica por el precio unitario y redondea a 2 decimales



***Ilustración 14 CalculatePriceSpent Recipe Models***

Cuando se elimina una fabricación, este método devuelve al inventario la cantidad gastada



***Ilustración 15 restoreStockInputs Recipe Models***

**Informe sobre el Front-end del módulo fabricación**

**CreateManufacturingModal.jsx**

Importaciones:

El React, useState, useEffect es un majeo de estados

El api es una instancia de Axios que nos sirve para hacer peticiones al backend

Swal es una librería que se utiliza para mostrar alertas bonitas

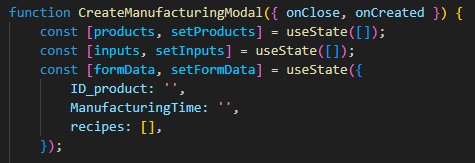


Estado del componente:

En las constantes products, inputs guardara mas adelante los valores correspondientes que el usuario registre

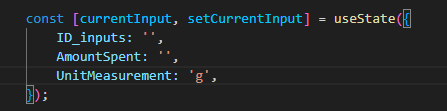
El FromData almacenara los datos que el usuario registre en los cuales tiene:

* ID\_Product; es el producto que se va a fabricar
* ManufacturingTime: el tiempo que tarda en hacerse.
* recipes: los insumos que usará ese producto.



Manejo de insumos temporales:

El usuario escoge un insumo, cantidad y unidad, al momento en el que presione el botón de agregar se pasa a la lista de fabricación, y esta lista se limpia cada ves que se agregue uno nuevo



Cargar datos:

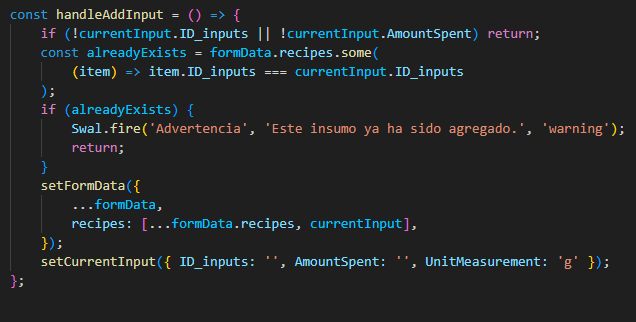
Al abrir el modal se carga los productos e insumos desde la API y quedan guardados en los estados products e inputs, que al momento de el usuario le da en el <select> los pueda ver

Texto

El contenido generado por IA puede ser incorrecto.

Agregar insumos:

Primero se valida de que haya insumos y una cantidad que no supere el stock y que no se agreguen números negativos, se verifica que el insumo no se repita por medio de Arrays y si todo el proceso se cumple correctamente, queda guardado en recipes dentro de la fromData



Eliminar insumos

Elimina el insumo en la posición indicada

Texto

El contenido generado por IA puede ser incorrecto.

Crear fabricación

Cuando se presiona el botón de crear fabricación, la información (formData) es enviada al backend, saltara una alerta de éxito, después llamara al onCreated() para que el componente padre actualice su lista y se cierra el modal

Texto

El contenido generado por IA puede ser incorrecto.

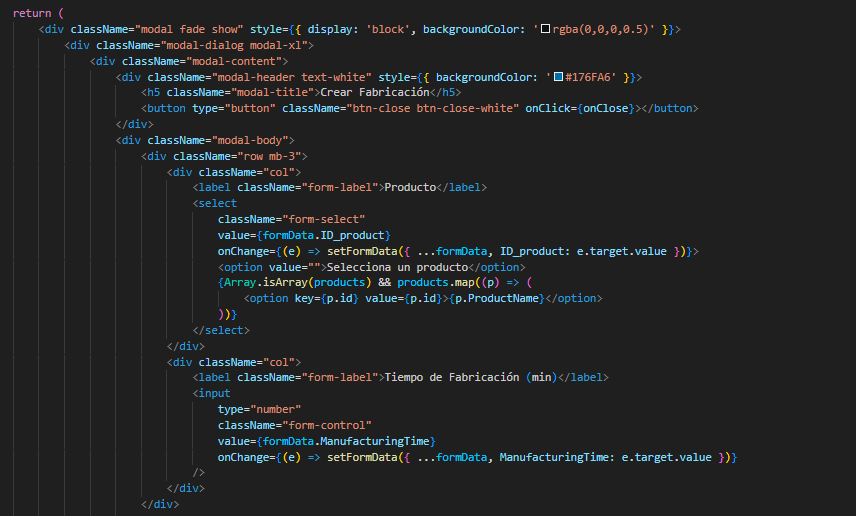
Estructura del modal

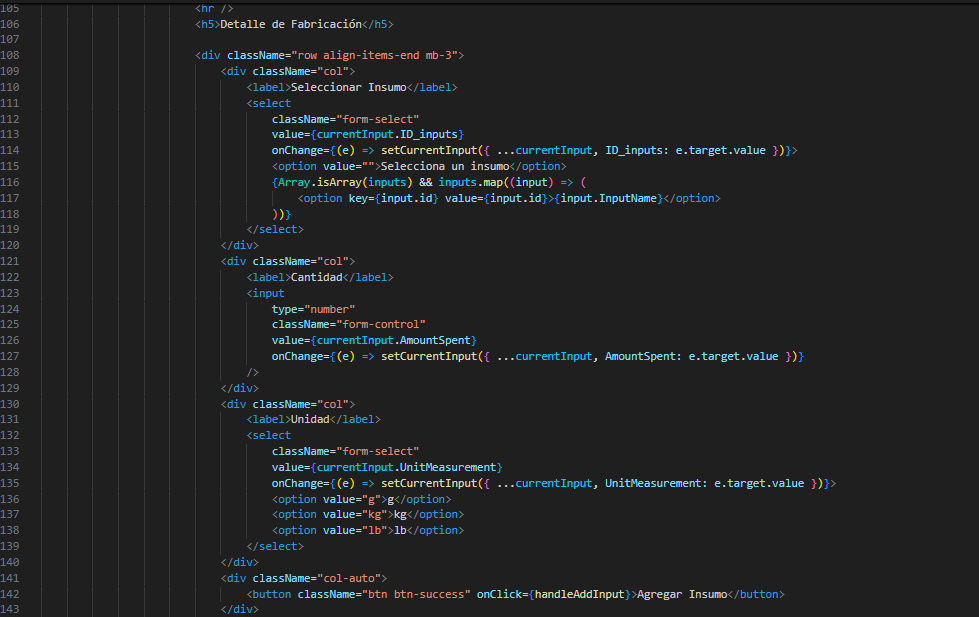
En este HTML y CSS este compuesto por tres secciones:

Header: se encuentra el titulo y el botón de cerrar

Body: está el formulario

Footer: botones para crear y eliminar





Texto

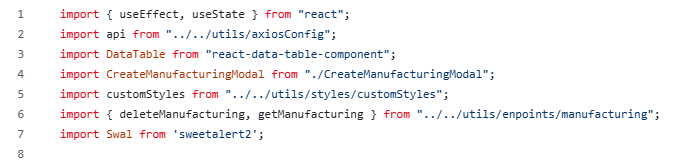
El contenido generado por IA puede ser incorrecto.

ManufacturingList.jsx

Importaciones:

Datatable: Liberia que muestra tablas interactivas

deleteManufacturing / getManufacturing: funciones que llaman al backend para borrar u obtener datos



Estados:

Se muestra la lista de fabricaciones que viene de la base de datos, y controla si mostrar o no el modal

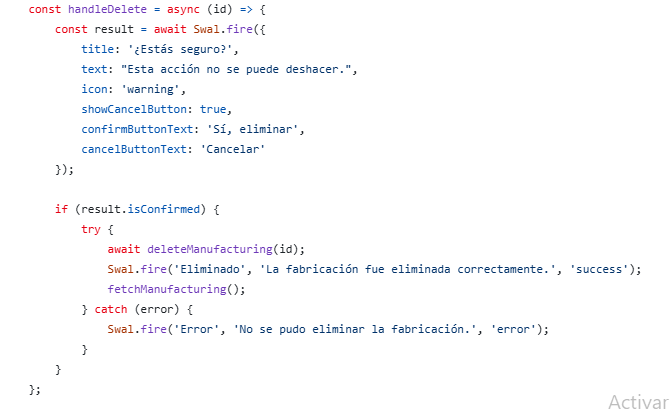
Además, apenas se abra el componente se llama de una ves el fetchManufacturing() que lo que hace es traer todos los datos del servidor

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

El contenido generado por IA puede ser incorrecto.

Eliminar una fabricación:

Se muestra un mensaje de que si esta seguro de eliminar, si lo confirmas llamara al deleteManufacturing(id) que se encuentra en el backend, dependiendo como se ejecute la solicitud se mostrara un mensaje de éxito o de error y por ultimo se actualiza la tabla llamando fetchManufacturing().



Obtener lista del backend:

Llama a getManufacturing() para obtener los registros del backend, y se actualiza el estado de manufacturing

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

El contenido generado por IA puede ser incorrecto.

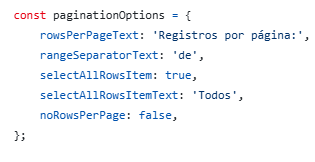
Columnas de la tabla

Es la vista de las columnas, en donde esta el título, los datos a mostrar, el orden en el que los registros aparecen, como se encuentra centrado con los botones de eliminar y editar





Opciones de pagina



Estructura de Manufacturig list

Aquí se encuentra todo lo que el usuario ve, como el título, el botón de crear fabricación, la tabla con la lista de fabricacion

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

Enpoints

Es un servicio que sirve como puente entre el frontend con react y el backend Api en laravel

Obtener lista de fabricantes

El api.get('/manufacturing') llama al backend para traer todas las fabricaciones, si la API responde bien se retorna al response.data que es la lista de las fabricaciones

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

El contenido generado por IA puede ser incorrecto.

Crear una nueva fabricación

El api.post("/manufacturing", data) envía los datos al backend para crear un nuevo registro, se retornará el objeto de la API devuelva en este caso es el nuevo registro que se creo

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

El contenido generado por IA puede ser incorrecto.

Eliminar una fabricación

El api.delete(/manufacturing/${id}) hace una petición DELETE a la API para borrar una fabricación especifica con el ID

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

El contenido generado por IA puede ser incorrecto.