

IMPLEMENTACIÓN DE UNA PLATAFORMA WEB PARA LA GESTIÓN DE LOS  
INVENTARIOS DE MATERIA PRIMA Y PRODUCTO TERMINADO EN LA PANADERÍA  
PAN DORADO LA 35 DE LA CIUDAD DE MEDELLÍN

DIEGO GALLEGO JIMÉNEZ

ESTIVENSON ESTRADA ALVAREZ

SEBASTIÁN CAMILO ARISMENDI ARIAS

UNIVERSIDAD CATÓLICA DE ORIENTE

FACULTAD DE INGENIERÍAS

PROGRAMA DE INGENIERÍA DE SISTEMAS

RIONEGRO ANTIOQUIA

2020

IMPLEMENTACIÓN DE UNA PLATAFORMA WEB PARA LA GESTIÓN DE LOS  
INVENTARIOS DE MATERIA PRIMA Y PRODUCTO TERMINADO EN LA PANADERÍA  
PAN DORADO LA 35 DE LA CIUDAD DE MEDELLÍN

DIEGO GALLEGO JIMÉNEZ

ESTIVENSON ESTRADA ALVAREZ

SEBASTIÁN CAMILO ARISMENDI ARIAS

Trabajo de grado para obtener el título de:

Ingeniero de sistemas

Asesor

Alejandro Ramírez Hernández

UNIVERSIDAD CATÓLICA DE ORIENTE

FACULTAD DE INGENIERÍAS

PROGRAMA DE INGENIERÍA DE SISTEMAS

RIONEGRO ANTIOQUIA

2020

Nota de aceptación:

---

---

---

---

Firma del presidente del jurado

---

Firma del jurado

---

Firma del jurado

Rionegro Julio de 2020

## CONTENIDO

ANTECEDENTES .....	7
1. PLANTEAMIENTO DEL PROBLEMA.....	9
2. JUSTIFICACIÓN .....	11
3. APORTES DE LA INGENIERÍA DE SISTEMAS A OTRAS ÁREAS DEL CONOCIMIENTO.....	12
4. OBJETIVOS.....	13
4.1 General.....	13
4.2 Específicos.....	13
5. MARCO TEÓRICO.....	14
Conceptos sobre el proceso para la elaboración de un producto final.....	14
Conceptos generales .....	15
Conceptos de desarrollo de software .....	16
6. DISEÑO METODOLÓGICO .....	21
6.1 Desarrollo e Implementación.....	22
Base de datos .....	22
Backend.....	23
FrontEnd.....	27
Seguridad .....	33

Seguridad del Backend .....	33
Seguridad de Frontend .....	34
Notificación de correos .....	36
7. RESULTADOS .....	37
7.1 Requisitos .....	39
7.2 Diseño del Backend .....	155
Capa de Dominio .....	155
Paquete src/main/java .....	156
Paquete JRE System Library .....	161
Maven Dependencies .....	162
Capa de Persistencia .....	163
Paquete src/main/resources .....	168
Paquete JRE System Library .....	170
Paquete Maven Dependencias .....	171
Capa Web .....	172
Paquete Java Resources .....	172
Paquete Java Script Resources .....	176
7.3 Diseño del Frontend .....	177
Paquete con los componentes .....	180
Carpeta de dominio .....	197

Carpeta de pipes.....	198
Carpeta service .....	199
7.4 Base de Datos .....	201
8. CONCLUSIONES .....	215
9. RECOMENDACIONES .....	217
10. REFERENCIAS BIBLIOGRÁFICAS.....	218

## ANTECEDENTES

Los productos de pastelería y panadería llegan a Colombia desde el siglo XVI, en la época de conquista,

*mediante el ejercicio de intercambio de comestibles entre españoles e indígenas, ingresando productos europeos a nuestra tierra. La evolución de los comestibles en Colombia fue lenta, inicialmente se conseguían rosquetas y bizcochos en las chicherías de la ciudad de Bogotá y solo a finales del siglo XIX, inauguraron las primeras empresas de este sector económico, como la Fábrica de galletas del Señor A. de J. Martínez, entre otros. (Muñoz, 2013)*

Investigando acerca del manejo de la materia prima, podemos observar que aproximadamente el 50% de las panaderías de Medellín y sus alrededores no tienen un control adecuado de los insumos que se utilizan para la elaboración de un producto. Entre los meses de septiembre y octubre hemos evaluado varias panaderías, entre ellas (el mundo del pan y Deguspan (Bello Ant)), entrevistando a sus dueños y/o administradores, concluimos que los procesos utilizados en la zona de producción están arrojando una manipulación inadecuada a la hora de elaborar las fórmulas establecidas, dado que los panaderos, pasteleros y reposteros no miden los porcentajes adecuados, evidenciando así desperdicios en cada producto de hasta 10 gr. de más.

Encontrando el problema, hemos llegado a que los sistemas para administrar internamente la empresa no son los apropiados para evitar aquellas pérdidas que pueden llevar a la panadería a crisis económicas. Hasta hace pocos años el control administrativo se realizaba con papel y lápiz, afortunadamente y gracias a la ciencia y a la tecnología, podemos encontrar en el mercado y de manera virtual, ciertos softwares que nos ayudan a administrar la materia prima,

pero no nos garantizan llevar una contabilidad 100% efectiva a la hora de pesar y realizar las masas madres, hablando de gramajes exactos.

Actualmente las panaderías están adquiriendo mejores métodos para contabilizar los gramajes exactos a la hora de la elaboración de los productos, de esta manera se obtendría una mejor rentabilidad. Gracias a la ciencia existen algunos softwares, que permiten llevar gran parte del control de inventarios, algunos son enfocados a la web y permiten ingresar desde cualquier lugar y verificar los insumos que se han consumido o están a punto de agotar, algunas páginas investigadas son: Mys inventarios, y Odoo. Al realizar el registro de usuario para interactuar, encuentro que no tienen la funcionalidad de contabilizar los gramajes exactos a la hora de pesar las fórmulas, tampoco tiene la posibilidad de tener un preaviso a la hora de tener inexistencia de insumo, esto para mejorar la eficiencia y control exacto de insumos tanto en pérdida como ganancia.

## 1. PLANTEAMIENTO DEL PROBLEMA

El negocio de la panadería y repostería en Medellín es un mercado muy competitivo, debido al gran número de panaderías que hay en la ciudad. Los consumidores a la hora de escoger una panadería buscan una muy buena atención, un lugar agradable y que el producto que consumen sea de gran calidad y al precio justo.

En los negocios de panaderías en la ciudad de Medellín se evidencia una problemática en común la pérdida de insumos. A la hora de preparar los mojes, se compara la cantidad de materia prima consumida a granel con la cantidad de productos terminados X mojes hechos y las pérdidas de producto son altas. Ya sea por el desarrollo de fórmulas (mojes) fuera de norma o por materiales que van sobrando y terminan en la basura. Ocasionando que el negocio por la cantidad de desperdicio no sea tan rentable como se espera y que la calidad del producto terminado tenga variaciones. Para que un negocio pueda sobresalir, necesita elaborar productos que cuente con gran calidad y de procesos controlados los cuales ayuden a tener estadísticas de cuánto material se consume, el desperdicio que este genera y las utilidades. Tener un estándar y un control de producción en cualquier empresa ayuda a tener un producto terminado con mayor calidad, a reducir costos y a ser más competitivos.

Panadería Pan Dorado la 33 de la ciudad de Medellín en pro de ser más competitiva y eficiente, busca un adecuado manejo de los insumos necesarios para la fabricación de los productos de panaderos, pasteleros y reposteros, ya que está generando pérdidas económicas del 3% al 15% de las ganancias obtenidas. Por consiguiente Teniendo un control sobre su inventario de materias primas y de productos terminados, con el cual estandarizar sus procesos, mantener sus desperdicios de materias primas por debajo del 3%, tener un control del consumo día, mes y año, de cada insumo y producto terminado y posicionarse en el mercado es uno de los objetivos,

se pretende desarrollar una plataforma web en la cual el gerente o dueño del local pueda verificar el gasto de insumos y la cantidad de productos fabricados a la hora de la producción.

## 2. JUSTIFICACIÓN

El inventario es uno de los activos más importantes en una panadería, sea cual sea el control que se lleve, ya que el manejo inadecuado o equivocado de este puede causar al negocio un exceso de mercancía o escasez o pérdida del mismo.

La automatización de los procesos de inventario en una empresa es un factor de vital importancia en la organización de esta. En la actualidad se ha observado una gran innovación y desarrollo de tecnologías de la información, que ha permitido manejar datos de forma descentralizada, solo con la implementación de plataformas web que nos permiten acceder a la información desde cualquier lugar y en todo momento solo es necesario tener una conexión a internet, estas plataformas son capaces de manejar sistemas de información en los diferentes sectores de la sociedad como económicos, educativos, políticos, sociales, entre otros.

La conveniencia de implementar una plataforma web que almacene y procese de una forma más efectiva la información de los procesos de inventarios, para lograr un control integral de los materiales que ingresan y que se utilizan. Tendrá un impacto positivo en cualquier negocio porque la contar con una información confiable, agilizando y facilitando el trabajo y control del material a utilizar permite que, al desarrollar un producto terminado, este se ajuste al presupuesto previsto para que sea rentable.

Los miembros de la comunidad se verán beneficiados de la implementación de esta plataforma web de automatización en el control de inventarios, porque al estandarizar procesos se logra obtener productos de mejor calidad, menor precio y que esta calidad sea estándar sin variaciones.

### 3. APORTES DE LA INGENIERÍA DE SISTEMAS A OTRAS ÁREAS DEL CONOCIMIENTO

La ingeniería de sistemas es un área que se puede aplicar a cualquier campo, para este caso la panadería. Actualmente encontramos panaderías con software para el manejo de ventas y control de inventarios, facilitando las actividades del personal. Este tipo de sistemas no sólo se aplica para panadería, la mayoría de las empresas tienen una o varias aplicaciones para sus ventas e inventarios, todo ello gracias al aporte de la informática.

Si bien, poder llevar a cabo un uso organizado para la panadería de los inventarios de materias primas, la elaboración de fórmulas, el control de los insumos y desperdicios al momento de preparar los productos finales es un aporte más que puede relacionar a las panaderías con la informática.

El aporte generado por parte de la ingeniería de sistemas en este proyecto demuestra que no solos puede contribuir a las panaderías con sistemas de ventas e inventarios, sino que se pueden realizar otro tipo de contribuciones cómo el control de desperdicios, manejo de fórmulas, inventarios de producción de los productos finales, entre otros.

Cada día, el campo de la tecnología crece exponencialmente para diferentes ramas de la industria, mejorando lo que se ha creado e innovando para el futuro, así que, para el campo de la panadería se hace la intención por medio de la informática y la ingeniería en ayudar a contribuir en aquellas cosas que aún no se han realizado.

## 4. OBJETIVOS

### 4.1 General.

Implementar una plataforma web para la gestión de los inventarios de materia prima y producto terminado en la panadería Pan Dorado la 35 de la ciudad de Medellín.

### 4.2 Específicos

- Identificar los requerimientos funcionales y no funcionales que debe tener la plataforma.
- Diseñar una arquitectura para la estructura, funcionalidad e integración del software.
- Construir una aplicación web para el manejo de fórmulas y control de inventarios de materia prima y producto terminado.

## 5. MARCO TEÓRICO

### Conceptos sobre el proceso para la elaboración de un producto final

El producto final hace referencia al resultado esperado de una producción de x producto de panadería, el cual es vendido y consumido por el cliente. Algunos de estos son: pastel, buñuelo, tortas, postres, panes, entre otros.

A continuación, algunos conceptos de panadería que son necesarios para la elaboración de un producto:

- Masa. Se trata del nombre que recibe la mezcla cuando ya está formada o amasada. Es lo más importante de la panadería.
- Amasar. Es dar forma a un trozo de masa, para armar el pan, generalmente la masa se estira y se arrolla y luego se da forma al pan que requiere, ya sea con queso o sin queso.
- Descansar La Masa. Es dejar reposar una masa el tiempo necesario para que afloje la tensión y nos permita darle la forma que deseamos sin que se reviente.
- Empaste o Hojaldre. Es pocas palabras es Grasa mezclada con Harina y es utilizada para elaborar pasta laminada.
- Levadura. Son hongos microscópicos responsables de la fermentación de diversos materiales orgánicos principalmente azúcares. Organismos vivos que rompe las moléculas de almidón en azúcares.
- Fermento. Se le llaman fermentos ya que son componentes de la levadura, organismos vivos que aceleran la formación de una masa.
- Harina. Es la que contiene todo el grano de trigo, menos el salvado y el germen.
- Molde. Recipiente o caja metálica o de otra materia que se utiliza para hornear ciertos panes.

- Moje. Se trata del nombre que recibe la masa cuando ya está amasada.
- Fórmula: Es la receta para preparar un producto de panadería. La fórmula se compone de los ingredientes, cantidad de ingredientes y observaciones que ayudan al panadero a preparar de una manera exitosa el producto final.

### Conceptos generales

Software: Podemos definir software como un conjunto de programas que se ejecutan en una computadora, celular, Tablet y otros dispositivos tecnológicos. Sin embargo, según el sitio web de Informática Milenium define software como:

*El Software son los programas de aplicación y los sistemas operativos que permiten que la computadora pueda desempeñar tareas inteligentes, dirigiendo a los componentes físicos o hardware con instrucciones y datos a través de diferentes tipos de programas.*

(informaticamilenium, s.f.)

Aplicación web: Una aplicación web es un programa informático construido y diseñado para usarse exclusivamente dentro de un navegador web tal como Google Chrome, Internet Explorer, entre otros. El cual accede a un servidor web a través de internet.

Servidor: Es una computadora destinada a recibir peticiones de computadoras, celulares, tablets, entre otros para devolver respuestas de manera oportuna a dichas peticiones. Estos tienen la capacidad de guardar y enviar información, distribuir tareas y tienen la peculiaridad de que deben permanecer de manera funcional en todo momento.

Git: Es un software de control de versiones diseñado para llevar orden y dominio de los cambios que se realizan sobre los elementos de un programa o código de una aplicación. Git permite hacer desarrollos de software de manera colaborativa y de una manera controlada.

Arquitectura de software: Es el diseño a más alto nivel de la estructura del software.

Lenguaje Interpretado: Es un lenguaje de programación que no se compila para poder ser ejecutado.

Lenguaje no interpretado: Es un lenguaje de programación que debe de ser compilado para poder ejecutarse.

Usuario: Es la persona que hace uso de la aplicación.

Trello: Es una herramienta para organizar y priorizar proyectos de forma colaborativa, haciendo uso de tarjetas, listas y tableros. Para este proyecto, se usó en la construcción de historias de usuarios.

Pixabay: Es un sitio web de imágenes, donde la comunidad comparte sus imágenes sin derechos de autor. Las imágenes usadas en la aplicación fueron tomadas de <https://pixabay.com/>.

### Conceptos de desarrollo de software

En el desarrollo de software las aplicaciones se componen por dos frentes, estos frentes son el BackEnd y el FrontEnd. Entre los conceptos del desarrollo de software por el lado del BackEnd se tienen los siguientes:

Backend: Se enfoca en hacer todo lo que está detrás de una página web para que esta funcione correctamente. El Backend “se encarga de realizar las peticiones a la base de datos para obtener la información, los cuales procesa para enviar al usuario, realiza la conexión al servidor de la base de datos y otras tareas que permiten funcionar la aplicación de manera satisfactoria.” (Benites, s.f)

Existen diferentes programas que permiten realizar un desarrollo de Backend, sin embargo, para el desarrollo este proyecto se hará uso de Java con algunos framework y de una base de datos MySql.

**Framework:** Un framework de desarrollo se define como una estructura de trabajo de soporte definida que permite una base organizada para realizar desarrollos de software. Los frameworks evitan realizar malas prácticas de programación a la hora de desarrollar software, alguno de estos beneficios es:

- Evitan escribir código repetitivo.
- Hacen más fácil y ligero el desarrollo de las aplicaciones.
- Usar buenas prácticas de programación.

**Spring boot:** Es una herramienta que brinda gran ayuda al desarrollador para realizar software. El Señor Ruben Pahino define Spring boot de la siguiente manera:

*Es un framework Open Source que facilita la creación de aplicaciones de todo tipo. Si bien es cierto que, por lo que es más conocido es por la inyección de dependencias, Spring Framework está dividido en diversos módulos que podemos utilizar, ofreciéndonos muchas más funcionalidades: Core container, Web, acceso a datos, programación orientada a objetos, Instrumentación y pruebas de código. (Pahino, 2020)*

**Maven:** Es una herramienta de automatización para la construcción del código. Maven es “es una herramienta open-source, que se creó en 2001 con el objetivo de simplificar los procesos de build (compilar y generar ejecutables a partir del código fuente).” (Garzas, 2014).

Para generar la comunicación entre el BackEnd y FrontEnd fue necesario exponer servicios, estos servicios son el puente que permiten la conexión entre ambos frentes. Para la implementación de este proyecto se usó API Rest.

**API Rest:** Es una interfaz de programa de aplicación, en sus siglas API. Esta permite utilizar solicitudes HTTP para consultar, insertar, publicar y eliminar datos. (Rouse, 2017) En su detalle son:

- Get: Para consultar
- Post: Para crear
- Put: Para editar
- Delete: Para eliminar

Json Web Token(JWT): Son un conjunto de medios de seguridad para el manejo de autenticación de aplicaciones web. Se basa en un Json que genera un token para poder enviar datos entre servicios.

Sendinblue: Es una empresa que ofrece diferentes soluciones de software para el marketing relacional. Para este proyecto se utilizó sendinblue, usando la opción de APICAMPAIN para el envío de correos electrónicos, este permite generar 300 email diarios de manera gratuita. La opción APICAMPAIN permite tener 4 campañas o plantillas de correos predestinados que están almacenados en la lista de sendinblue.

Arquitectura en tres capas: Es un tipo de arquitectura de software, la cual permite al sistema dividirse en tres capas, segregando funcionalidades y bajo acoplamiento en la aplicación.

Las tres capas son:

- Persistencia: Capa encargada de guardar la información, es la encargada de realizar las operaciones sobre la base de datos.
- Negocio: Es la capa que tiene la lógica de la aplicación. Esta está conectada a la capa de persistencia.
- Presentación: Es la capa del usuario. Esta es la que permite que el usuario interactúe con el sistema y de enviar dichas acciones a la capa de negocio.

Java: Es un lenguaje de programación gratuito, usado para desarrollar software por el lado de backend.

IDE: Es una herramienta que permite el desarrollo de aplicaciones. Las siglas IDE significan Entorno de desarrollo integrado.

Eclipse Jee 2019-06: Es un IDE el cual para ejecutar y desarrollar aplicaciones Java, existen diferentes versiones de un IDE, sin embargo, para este proyecto se usó el Jee 2019-06.

MySQL: Es un motor para la gestión de base de datos relacionales, su principal objetivo es almacenar datos.

MySQL Workbench: Es una herramienta visual para la gestión de las bases de datos de MySQL.

Modelo Entidad Relación(MER): Es un diagrama que representa las relaciones entre las tablas de una base de datos.

Java Persistence API(JPA): Es un framework que permite interactuar con la base de datos por medio de objetos java.

Dozer: Es un framework que se especializa en la definición de un tipo de conversión.

Se hizo mención a los conceptos de BackEnd, ahora se darán a conocer algunos conceptos de FrontEnd:

FrontEnd: Es todo lo que el usuario puede ver e interactuar con la aplicación. Es la parte visual, la cual se construye a partir de diferentes herramientas y framework con el fin de ofrecer al usuario una buena usabilidad.

Servidor web: Un servidor web es un software utilizado para almacenar información web que provee a los dispositivos de usuarios como computadoras y celulares.

Angular: Es un framework de código abierto creado por Google para desarrollar aplicaciones web. Angular permite separar el FrontEnd y el BackEnd de la aplicación con el fin de ayudar al programador a realizar un software más limpio y con buenas prácticas.

Angular CLI: Es una herramienta de línea de comando que permite crear proyectos web con los archivos necesarios para que la aplicación funcione correctamente.

Angular 7: Es la versión del framework Angular, el cual se usará para el desarrollo del FrontEnd.

HTML: Es un lenguaje de programación no interpretado con el que se construye el contenido de páginas web. “Básicamente se trata de un conjunto de etiquetas que sirven para definir el texto y otros elementos que compondrán una página web, como imágenes, listas, vídeos, etc.” (Alvarez, 2001) Las siglas HTML significan “HyperText Markup Language”.

HTML 5: Es la versión número 5 de la revisión de la tecnología HTML.

CSS: Es un lenguaje que permite dar apariencia, color y estilos a los contenidos HTML de las páginas web. Las siglas significan “Cascading Style Sheets”, en español “Hojas de estilo en cascada”.

Bootstrap: Es un framework que permite crear interfaces web con CSS y JavaScript. Este permite crear diseños adaptables al tamaño de la pantalla de cualquier dispositivo, además ofrece gran variedad de estilos y diseños para el desarrollo de la aplicación del lado del FrontEnd, como botones, tablas, textos, menús, ventanas emergentes, entre otros.

JavaScript: Es un lenguaje de programación no interpretado que se usa para el desarrollo web, este es uno de los más importantes de la actualidad por su utilidad y facilidad de usar para dar dinamismo a las páginas web. Es un lenguaje orientado a objetos.

TypeScript: Es un lenguaje de programación de código abierto desarrollado por Microsoft, tiene la característica de ser orientado a objetos. Este permite realizar desarrollos del lado del FrontEnd de la aplicación.

Diseño Web Adaptable: Es una técnica de diseño y desarrollo web para que las páginas web se adapten a cualquier tamaño de pantalla del dispositivo.

## 6. DISEÑO METODOLÓGICO

Lo primero que se hizo fue una investigación sobre la panadería con el fin de conocer cómo es el proceso completo para la elaboración de un producto. Esta investigación permitió conocer el proceso y los conceptos claves para el entendimiento del negocio. Se habló con el experto de la panadería, el cual dio a conocer cada detalle necesario para la elaboración de la aplicación.

Con el contexto del negocio y la problemática entendida, se realizó el levantamiento de requerimientos funcionales y no funcionales, los modelos entidad relación y la arquitectura a implementar para la aplicación. Después de obtener esto se inició con la creación de la base de datos relacional en MySQL, se creó cada tabla definida del MER y se ingresaron datos de prueba.

Luego, se empezó a desarrollar el software, y para ello se creó el proyecto del BackEnd como una aplicación de Spring Boot en Java; Se implementó

su respectiva capa de domino, persistencia y web. En la capa de Dominio se crean los dominios, en la capa de la persistencia se crea el modelo de datos con JPA para acceder a la base de datos de la aplicación, así mismo se crea la capa web para los controladores y servicios Rest que serán consumidos por el FrontEnd.

Después de esto, se inició con el proyecto del FrontEnd, donde lo primero fue la creación de las entidades de dominio, componentes y servicios. En los componentes se desarrolla cada vista requerida y los diseños para interactuar con la aplicación. Finalmente se realizó toda la

seguridad de la aplicación, desde el acceso a la aplicación, encriptación de datos, permisos de funcionalidades de acuerdo con el perfil de empleado y protección de rutas de la aplicación. Se utilizan varias tecnologías expuestas en el marco teórico para un desarrollo con buenas prácticas, patrones de diseño y experiencia de usuario.

Se utilizó la herramienta de trello para la construcción de las Historias de usuarios, las cuales fueron distribuidas en diez Sprints, donde cada Sprint duró aproximadamente un mes.

### 6.1 Desarrollo e Implementación

A continuación, se dará a conocer como fue el proceso para la construcción de la parte de BackEnd del software, teniendo en cuenta las definiciones presentadas en ítems anteriores.

#### *Base de datos*

Se inició con el bosquejo del MER, teniendo en cuenta las tablas para almacenar la información, las relaciones de tipo primarias y foráneas, los atributos que definirían cada tabla con sus respectivo tipo de dato, además, se definió cuando un campo de alguna tabla no puede estar vacío. Cuando se realizó el bosquejo, se procedió con la implementación y construcción de la base de datos relacional, para ello se usó MySQL Workbench, donde se implementó cada tabla definida, con sus respectivas claves primarias y foráneas, atributos y relaciones. Teniendo el MER, se creó la base de datos.

MySQL Workbench tiene la característica de crear la base de datos de una forma muy simple, este la construye a partir del MER, con la opción “Forward Engineer”. MySQL Workbench crea los insert de cada tabla, con los nombres de los campos, claves primarias ,calves foráneas, es decir, con cada una de las características definidas desde el MER.

Al tener la base de datos lista, se procedió con el ingreso de datos a cada una las tablas y se realizaron consultas a la base de datos con el fin de probar que la base de datos estuviese

correcta. Esto permitió corregir algunas relaciones y analizar posibles mejoras que no se contemplaron desde un inicio.

### *Backend*

El backend se inició con la definición de la arquitectura, no se podía iniciar con el desarrollo sin tener esto totalmente claro. Se definió una arquitectura de tres capas, la capa de dominio, la capa de persistencia y la capa web. Luego, se procedió con el desarrollo del backend, donde el lenguaje de programación usado fue Java, ya que se consideró que es el lenguaje más usado en la industria laboral y el que más daría utilidad y facilidad para el desarrollo del software.

para ellos se crearon tres proyectos de Spring boot tipo Maven, donde un proyecto fue la capa de dominio, otro fue la capa de persistencia, y por último el de la capa web. Los proyectos se generaron desde la página de <https://start.spring.io/>; Esta página permite crear de una manera muy sencilla la estructura base o el esqueleto de un proyecto tipo Maven.

Con el esqueleto del proyecto construido, se procedió con la comunicación de la base de datos con el backend, para ellos se configuró los parámetros necesarios en el archivo 'application.properties' de la capa web. A este archivo pueden acceder de otras partes de la aplicación, es decir, la capa de dominio y la capa web pueden acceder a esta configuración.

En cada capa, existe el archivo pom.xml, aquí se configuró las dependencias del sistema Java, las cuales vienen predefinidas. Esto fue gracias [start.spring.io](https://start.spring.io/) que construyó un esqueleto con todos los artefactos necesarios. En este archivo se realizó la comunicación entre las capas, poniendo cada capa como dependencias de las otras.

Después, En la capa de dominio, se realizó los modelos de dominio. Cada clase definida en el dominio hizo referencia a cada tabla que se implementó en la base de datos. Las entidades de dominio fueron las siguientes:

- Compra
- CompraDetalle
- CompraDetallePK
- Contraseña
- Empleado
- Empresa
- EmpresaSede
- Formula
- Ingrediente
- IngredientePK
- InventarioMateriaPrima
- Moje
- Persona
- Producto
- ProductoFinal
- Proveedor
- ProveedorTelefono
- Role
- Telefono
- TipoDocumento

- TipoPersona
- TipoTelefono
- UnidadMedida

Estas son las entidades definidas y que fueron usadas para el dominio, las cuales van a la base de datos, por medio de otras entidades que se expondrán más adelante. Además, se crearon otros dos paquetes, uno para las interfaces que tienen los métodos generales que pueden ser usados desde los controladores, llamado 'domain.repository' y otro denominado 'domain.service' para los métodos básicos de los servicios, que se usaron para el correcto funcionamiento del CRUD

En la capa de la persistencia, se implementó el desarrollo para poder interactuar con la base de datos por medio de objetos java, se crearon las entidades del dominio por medio de JPA. También se realizó la implementación de las interfaces que se usaron para las consultas a la base de datos, allí se pusieron todos los querys necesarios para traer la información.

Luego, se implantaron las inyecciones de dependencias, las cuales sirvieron para evitar código repetitivo y crear instancias para ciertas clases que están en la persistencia.

Después de esto, se inició el acceso a la base de datos, ya configurado el archivo 'application.properties', los modelos del dominio y las entidades de dominio con JPA, se pudo ingresar objetos a la base de datos. Además, se implementó una clase de casteo utilizando Dozer, con el propósito de evitar problemas en la conversión de estructuras y datos con el manejo de los objetos que representan las tablas de la base de datos.

Luego, se procedió con la implementación de los controladores Rest, encargados de exponer los servicios Rest. Esto se hizo en la capa web. Con esto, permitió empezar con el desarrollo de los CRUD del software. A continuación, los controladores implementados:

- CompraDetalleRestController
- CompraRestController
- ContraseñaRestController
- EmpleadoRestController
- EmpresaRestController
- EmpresaSedeController
- FormulaRestController
- IngredienteRestController
- InventarioMateriaPrimaRestController
- MojeRestController
- PersonaRestController
- ProductoRestController
- ProductoFinalRestController
- ProveedorRestController
- RoleRestController
- TelefonoRestController
- TipoDocumentoRestController
- TipoPersonaRestController
- TipoTelefonoRestController
- UnidadMedidaRestController

Cada uno de estos controladores se les creó los servicios Rest correspondientes para el CRUD, es decir, servicios para:

- Crear: Estos servicios fueron tipo POST.

- Eliminar: Estos servicios fueron tipo DELETE.
- Actualizar: Estos servicios fueron tipo POST.
- Consultar: Estos servicios fueron tipo GET.

Además, a medida que se fue construyendo el backend del software se fueron implementando otros servicios para el correcto funcionamiento de la aplicación. Y con las librerías que ofrece el framework de Spring, se expusieron los servicios para ser consumidos por el frontend.

Luego, se procedió con las pruebas de los servicios Rest. Cada servicio Rest fue probado con Postman, una aplicación que permite realizar envío de peticiones HTTP REST, con el fin de verificar que cada servicio retornara la información correcta.

### *FrontEnd*

Se inició creando el proyecto de frontend, para esto se usó la herramienta de angular CLI usando la consola de administrador de paquetes npm, los pasos para crear el proyecto fueron:

- `npm install @angular/cli`: Permite instalar angular cli.
- `npm install -g @angular/cli@latest`: Comando que permite instalar el framework de angular, para este proyecto se usó angular 7.
- `ng new nombre_de_mi_app`: Comando que permite crear el proyecto, para este proyecto se le dio el nombre de frontend.

Con estos pasos angular cli se encarga de generar todos los archivos necesarios para que el proyecto quede funcional. Después de tener el esqueleto del proyecto se inició con el desarrollo del software para el frontend, en primera instancia se implementó las clases de dominio, estas clases contienen los atributos con su respectivo tipo de datos. A continuación, las clases de dominio:

- compra.ts
- compra-detalle.ts
- compra-detalle-pk.ts
- empleado.ts
- empresa.ts
- formula.ts
- ingrediente.ts
- ingredient-pk.ts
- inventrario-materia-prima.ts
- moje.ts
- persona.ts
- producto.ts
- producto-final.ts
- proveedor.ts
- proveedor-telefono.ts
- rol.ts
- telefonoPersona.ts
- tipoDocumento.ts
- tipoPersona.ts
- tipoTelefono.ts
- unidad-medida.ts
- usuario.ts

Cada clase de dominio queda almacenada en una carpeta llamada 'domain', asignada para almacenar los dominios de la aplicación.

Teniendo el dominio, se realizó la implementación de los servicios, estos servicios corresponden a cada uno de los servicios rest que fueron expuesto desde el backend. Cada clase de los servicios contiene el endPoint que comunica al frontEnd con el backend, con estos endPoint se permitió al front consumir los servicios que fueron expuestos por el backend.

Luego de tener estos endPoint, se generaron los servicios para el CRUD, para crear, actualiza, eliminar y consultar. Así mismo, se generaron otros servicios que fueron necesarios para cumplir con las funcionalidades que requiere la aplicación para el correcto funcionamiento.

Cada clase para los servicios se creó con la consola del editor con el siguiente comando: 'ng generate service <nombre del servicio>'. Cada uno de ellos se crea por forma automática dentro de la carpeta de 'Service'.

Después se procedió con la creación de los componentes, donde se creó un componente para cada objeto de dominio que se encuentra en el proyecto. Cada componente contiene cuatro partes, estas son:

- CSS: Se usa para dar estilos a la vista de los componentes. Esto se hace sobre las etiquetas html.
- HTML: Con este se construye la estructura que define la parte visual de la vista.
- TypeScript: Es la parte permite construir la lógica y las funciones que tiene que realizar la vista.
- Pruebas de TypeScript: Permiten realizar pruebas a lo desarrollado en la lógica del TypeScript.

Estos componentes se crearon usando la consola de editor, con el comando ‘ng generate component <nombre del componente>’. Cada componente se crea de forma automática dentro de una carpeta a la cual se le asigna el nombre que le asignó en el comando de la consola. Cada componente contiene un path o una ruta que lo identifica para direccionar un evento, este evento es el que determina la funcionalidad a la hora de utilizar la aplicación, para esto se implementó un ‘app-routing.module’, este tiene la funcionalidad de definir las rutas de los componentes y direccionarlos para que el usuario pueda usar la aplicación. Se usó el framework de Angular 7 para la construcción de las funcionalidades de las vistas haciendo uso del typescript, html y css de los componentes.

A continuación, se hablará de cómo se desarrollaron estos componentes, de una forma más detallada, sin embargo, se mencionará lo principal. Se empezó desarrollando el módulo de talento humano, el cual se hizo en tres partes:

- **Personas:** En este módulo se generó el desarrollo para consultar, crear, actualizar y eliminar personas.
- **Empleados:** En este módulo se hizo el desarrollo para que una persona fuese un empleado, permitiendo consultar, actualizar y eliminar la información de un empleado. Además, implementó el desarrollo para generar las credenciales a un empleado, para que este puede ingresar a la aplicación, el detalle de esto se explicará mas adelante.
- **Utilidades:** En este módulo se hizo el desarrollo todo lo referente a los tipos de teléfonos, tipos de personas, tipos de documentos y roles. Se hizo como un módulo paramétrico donde el usuario puede determinar qué tipo de información usar en la aplicación. Es decir, se puede crear el tipo de teléfono o documento que él desee.

Después de tener el módulo de talento humano, se inició el desarrollo del módulo de compras e inventario. Este se realizó en tres partes:

- Compras: En este módulo se hizo el desarrollo para que el usuario pueda ingresar compras, las cuales alimentan el inventario de materias primas. Se generaron funcionalidades para crear una Compra, visualizar la información de una compra y actualizar la información de una compra.
  - Inventarios de materias primas: En este módulo se realizó la lógica necesaria para permitir al usuario consultar las materias primas que hay actualmente. Además, se hizo una vista para observar solo el inventario de las materias primas que están por debajo de la cantidad mínima permitida en el inventario.
  - Proveedores: En este módulo se hizo el desarrollo para consultar, ingresar, eliminar ya actualizar la información de los proveedores de insumos de la materia prima.
- Luego, de la finalización de compras e inventarios, se inició con el módulo de producción,

este módulo se construyó por tres partes:

- Fórmulas: En este módulo se hizo toda la implementación y lógica para la creación de las fórmulas que se usan para la preparación de un producto final. También, se hizo las vistas para actualizar, ingresar ingredientes y ver la información de las fórmulas.
- Mojes: En este módulo se realizó el desarrollo para crear los mojes referentes a un producto por medio de la selección de una fórmula. Además, la vista para observar el inventario de los mojes realizados.
- Productos terminados: En este módulo se hizo el desarrollo para terminar un producto y para determinar si un producto generó pérdida. Además, se implementaron las vistas para visualizar la información de los productos terminados.

Después se implementó los filtros de búsqueda de las vistas que lo requerían, para esto se usaron ‘pipes’, estos son almacenados en una carpeta denominada pipes, y se crearon por medio de la consola de editor con el siguiente comando ‘ng generate pipe <nombre del pipe>’. Estos llevan la lógica para permitir traer información de una manera fácil y rápida con base a lo que se necesite, en este caso se usó para generar filtros con diferentes características, lo cual permitió buscar la información más fácil al usuario.

También, se desarrolló la vista y la lógica de empresas, con su respectivo componente, allí se implementó la vista para ver la información y actualizar la información de una empresa. Luego de esto se procedió con la implementación de la vista y la lógica para el módulo de sedes de una empresa. En este módulo se creó las vistas para observar la información de la sede y editar la información de una sede.

Luego, se crearon los componentes para generar el navbar (barra de menú) y el footer (pie de página que va al final de cada vista). El navbar se hizo con accesos directos al módulo de sedes, producción, compras e inventario, talento humano y la opción para entrar al perfil del usuario (se explicará más adelante). En el esqueleto del proyecto hay un archivo llamado ‘app.component.html’ el cual hace la función de definir las etiquetas que mostrará siempre la aplicación, es decir, aquellos componentes que siempre se mostrarán en las vistas de la aplicación, para evitar generar código repetitivo en los componentes creados anteriormente. La estructura del archivo ‘app.component.html’ se compone por un header, un body y un footer, estas tres partes son las que siempre se mostrarán al usuario. Donde el header hace referencia al navbar, el body a los componentes que se crearon en función al dominio, y el footer al footer creado anteriormente.

### *Seguridad*

Después de tener todas las vistas de la aplicación y módulos ya funcionales se procedió con la implementación de la seguridad. Esta seguridad se hizo a nivel de backend y frontend, usando JWT (Jason web token). Básicamente consiste en generar un Token de seguridad válido para cada petición que se le realice desde el frontend al backend, si este token no se genera no permite ingresar a la aplicación, o en caso de estar en la aplicación ya navegando esta deja de funcionar. El token hace el papel de una llave electrónica para acceder a las funciones de la aplicación.

### *Seguridad del Backend*

Por el lado del backend, se inició instaló al proyecto la librería de 'io.jsonwebtoken' en las dependencias del archivo pom.xml del proyecto de Maven. Al instalar la dependencia, se implementaron en la capa web cuatro clases java, las clases fueron las siguientes:

- `wtAuthenticationEntryPoint`: Esta clase se hizo para el manejo de errores, por ejemplo: cuando un token no es válido, por ende, no hay permisos para ingresar a la aplicación.
- `JwtRequestFilter`: En esta clase se realizó la implementación para configurar la autenticación, donde se valida que el usuario este autenticado, y de no estar autenticado la seguridad de Spring no permitirá proseguir.
- `JwtTokenUtil`: En esta clase se hizo el desarrollo para crear el token, donde se define las características del token como: Fecha de caducidad del token, fecha de creación del token, el usuario, el rol, la generación de clave secreta usando el algoritmo de HS512 para la encriptación y otros campos que fueron necesarios.

- **WebSecurityConfig:** En esta clase se implementó la lógica para que el sistema sepa de donde cargar la encriptación de las contraseñas y validar que sean las mismas que se están ingresando a la aplicación.

Después de realizar estas clases, se editó la lógica en el CRUD de los servicios rest de la clase 'ContrasenaRestController', para que encripte la contraseña al crear las credenciales de un empleado. Además, se hizo un método para la autenticación de las credenciales que se están enviando desde el frontend para ingresar a la aplicación. Esto se hizo realizando inyección de dependencias de las clases de JwtTokenUtil y clases que traen las librerías de Spring Security cómo 'AuthenticationManager'.

### *Seguridad de Frontend*

Para la seguridad del frontend se inició con la instalación de la librería de 'jwt-decode', se hizo por medio de la consola de editor con el comando 'npm install --save jwt-decode'. Esta librería permite desencriptar el token y manipular la información que vienen dentro, y otras funcionalidades para trabajar con JWT.

Después de la instalación, se crearon cuatro servicios para la seguridad, estos servicios son:

- **AuthGaurdService:** Es un servicio que hace el papel de guardián, protege las rutas de los componentes que se encuentran en el 'app-routing.nodule.ts' evitando el ingreso a una de estas rutas si no hay una autenticación previa.
- **AuthGuarRoleService:** Es un servicio que se realizó para la segregación de funciones de la aplicación dependiendo del rol del empleado. Este protege el acceso a las rutas de un empleado que no tenga los permisos para acceder a él.

- `AuthenticationService`: En este servicio se hizo la lógica para obtener el token que envía el backend y se hace uso de él para verificar en todo momento que el usuario se encuentre autenticado en la aplicación.
- `BasicAuthHttpInterceptorService`: Este servicio se hizo para validar que el nombre de usuario y el token sean correctos. con el fin de enviar la autorización de acceso.

Teniendo estos servicios construidos, se inició con la implementación de la vista y la lógica del login. Desde la vista se envía el nombre de usuario y la contraseña, estos van al backend y se valida que este usuario y contraseñas coincidan con los datos almacenados en la base de datos, si las credenciales son verdaderas, el backend devuelve un token válido; este token es recibido por el frontend para permitir dar acceso al usuario. De no ser las credenciales correctas, no permite el ingreso a la aplicación.

Después, se crea la funcionalidad de asignar credenciales a un empleado, para esto, se editó la lógica del componente de empleados. En EL typescript se genera la lógica para crear de manera automática el nombre del usuario y la contraseña. En la vista que permite ver la información de un empleado, se implementó un botón que asigna las credenciales a el empleado seleccionado. Al crear estas credenciales, el empleado puede hacer uso de aplicación, ingresando con el usuario y contraseña que se asignó.

Luego, se realizó la implementación en el mismo componente para recuperar la contraseña de un empleado, en caso de haberla olvidado. También, se agregó la opción de consultar la información del perfil del empleado que se encuentra navegando en la aplicación, esto se hizo en el navbar.

Finalmente, se crea la funcionalidad para actualizar la contraseña del empleado.

### *Notificación de correos*

Para la implementación de notificación de correos, se usó la librería del api llamada *sendinblue*. Lo primero que se hizo fue instalar las dependencias en el archivo 'pom.xml' de la librería 'com.sendinblue' en el proyecto. Teniendo la dependencia instalada se procede con la creación de una cuenta en *Sedinblue* para poder hacer uso de ella. Luego se crearon dos campañas, una para el envío de correos de inventario y otra campaña para el envío de credenciales de usuario. Una campaña básicamente es la configuración de envíos de correos el cual genera una ruta o un endPoint que permite la conexión de nuestro proyecto con la API. Con esto realizado, se procede con el desarrollo a nivel de backend para el envío de correos, donde se implementó el desarrollo para tomar y guardas los correos de los empleados que requieren la notificación. Luego, se hizo la implementación para instanciar la conexión a *sendinblue*. Después, se construye el cuerpo del correo con su estructura de mensaje, los correos se componen de un usuario de envío, un usuario destinatario (correo de empelados), un asunto y un cuerpo de correo.

## 7. RESULTADOS

Objetivo 1: Identificar los requerimientos funcionales y no funcionales que debe tener la plataforma.

- Se logró entender el proceso de la panadería hasta la entrega del producto final al cliente.
- Hubo una comprensión correcta de los conceptos de negocio de la panadería.
- Se entendió el proceso para la creación de fórmulas con sus respectivos ingredientes.
- Al entender la panadería en general, se logró identificar las funcionalidades que necesitó la aplicación para dar solución al problema.

Objetivo 2: Diseñar una arquitectura para la estructura, funcionalidad e integración del software.

- Una capa de presentación (frontend), en la que el usuario puede interactuar con la aplicación.
- Una capa de aplicación(backend), donde se obtiene la información para cada una de las vistas de la presentación.
- Una capa de base de datos, donde se aloja y se inserta toda la información, distribuida en diferentes tablas que se usan para el correcto funcionamiento de la aplicación.

Objetivo 3: Construir una aplicación web para el manejo de fórmulas y control de inventarios de materia prima y producto terminado.

- Una aplicación con tres módulos principales: El de producción, talento humano y el de compras e inventario.

- Un módulo de producción, donde se crea y se consultan fórmulas de un producto.
- Un módulo de compras, donde se ingresan los insumos o ingredientes que se compran para la elaboración de los productos.
- Un módulo de inventario, donde se consultan los insumos de la panadería.
- Un módulo de inventario bajo para consultar aquellos insumos o materias primas que se están agotando.
- Notificación de correos cuando se agote un insumo del inventario. Y dos notificaciones automáticas de correo del inventario de los insumos, uno al inicio de jornada y otro al finalizar la jornada.
- Módulo para crear mojes de una formula y generar un producto final.
- Se obtiene el porcentaje de pérdidas de la producción de un producto final, permitiendo ver la rentabilidad de la producción de dicho producto.
- Se obtiene el valor del costo de una fórmula de un producto.
- Módulo de talento humano para administrar las personas y empelados de la panadería. Permitiendo asignar un rol a los empleados y las credenciales de ingreso para hacer uso de la aplicación.

## 7.1 Requisitos

## Sprint 1

- HU 1:

The screenshot shows a Trello card with the following content:

- Title:** HU-1. Instalación de programas para el desarrollo del proyecto. (with a close button 'X')
- Location:** en la lista [Sprint 1](#)
- Description:** Yo como desarrollador requiero la instalación de los programas necesarios para la construcción del proyecto. Los programas requeridos son los siguientes:
  1. Eclipse IDE 2019-06
  3. Instalación del motor de base de datos MySQL Workbench
  4. Instalación del editor de texto Visual estudio Code
- Acceptance Criteria:** Criterios de aceptación (with buttons: Ocultar elementos completados, Eliminar). A progress bar shows 100% completion.
  - El IDE de Eclipse debe de quedar funcional.
  - El motor de base de datos debe de quedar configurado correctamente y listo para usarse.
  - El editor de texto debe de quedar instalado correctamente listo para usarse.(with a button: Añada un elemento)
- Right Sidebar:**
  - AÑADIR A LA TARJETA:** Miembros, Etiquetas, Checklist, Vencimiento, Adjunto, Portada.
  - POWER-UPS:** Consequir Power-Ups. Consiga Power-Ups ilimitados y mucho más. Actualizar el equipo.
  - ACCIONES:** (empty)

Figura 7.1.1

Autor: Auditoría propia, herramienta usada: Trello.

- HU 2

**HU-2. Implementar modelo entidad relación de la base de datos.**  
en la lista [Sprint 1](#)

**Descripción** Editar

Yo como desarrollador requiero diseñar y crear el modelo entidad relación(MER) de la aplicación para crear la base de datos relacional.

**Criterios de aceptación** Ocultar elementos completados Eliminar

100%

- El modelo se debe de crear en MySQL Workbench con extensión .mwb
- Se debe de crear también el MER en un archivo .PDF
- Se deben de visualizar el nombre de las tablas, los atributos de cada tabla, el tipo de dato del atributo, el tamaño del atributo y las relaciones entre las tablas.

Añada un elemento

**AÑADIR A LA TARJETA**

- Miembros
- Etiquetas
- Checklist
- Vencimiento
- Adjunto
- Portada

**POWER-UPS**

Conseguir Power-Ups

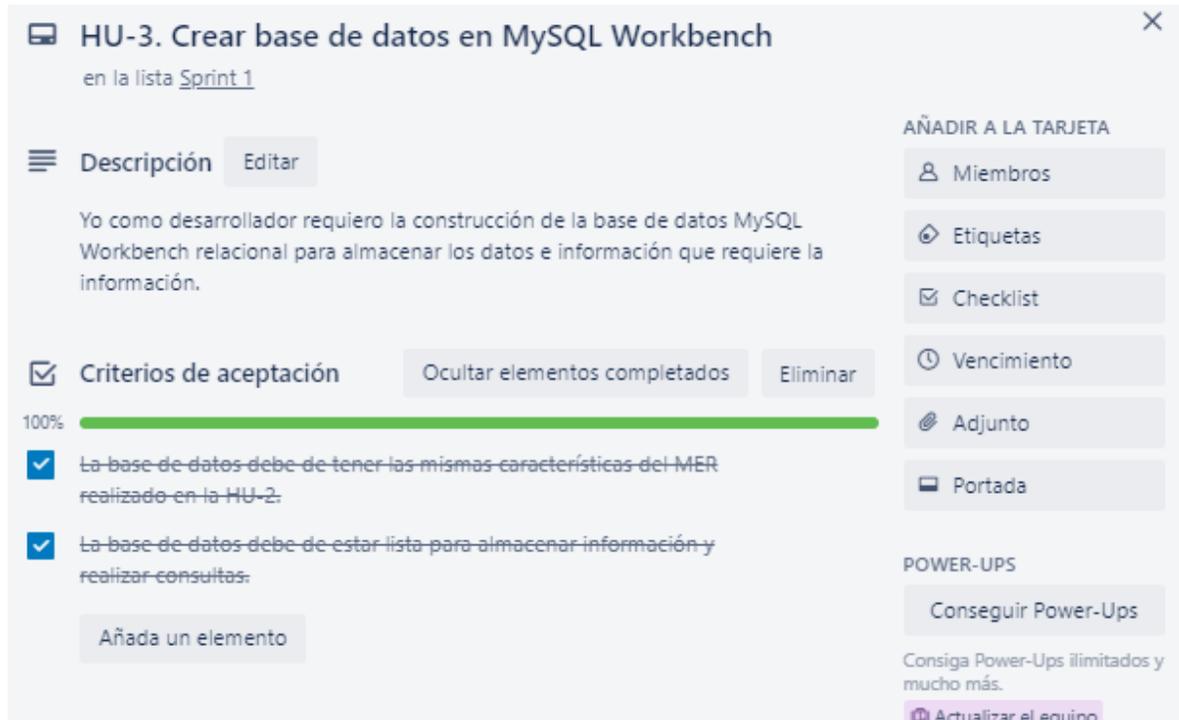
Consiga Power-Ups ilimitados y mucho más.

Actualizar el equipo

Figura 7.1.2

Autor: Auditoría propia, herramienta usada: Trello.

- HU 3



The image shows a Trello card with the following content:

- Title:** HU-3. Crear base de datos en MySQL Workbench
- Location:** en la lista [Sprint 1](#)
- Description:** Yo como desarrollador requiero la construcción de la base de datos MySQL Workbench relacional para almacenar los datos e información que requiere la información.
- Acceptance Criteria:** 100% complete. Two criteria are listed and checked: "La base de datos debe de tener las mismas características del MER realizado en la HU-2:" and "La base de datos debe de estar lista para almacenar información y realizar consultas:".
- Right Sidebar:** Includes sections for "AÑADIR A LA TARJETA" (Members, Labels, Checklist, Due Date, Attachment, Cover) and "POWER-UPS" (Get Power-Ups, Upgrade team).

Figura 7.1.3

Autor: Auditoría propia, herramienta usada: Trello.

- HU 4

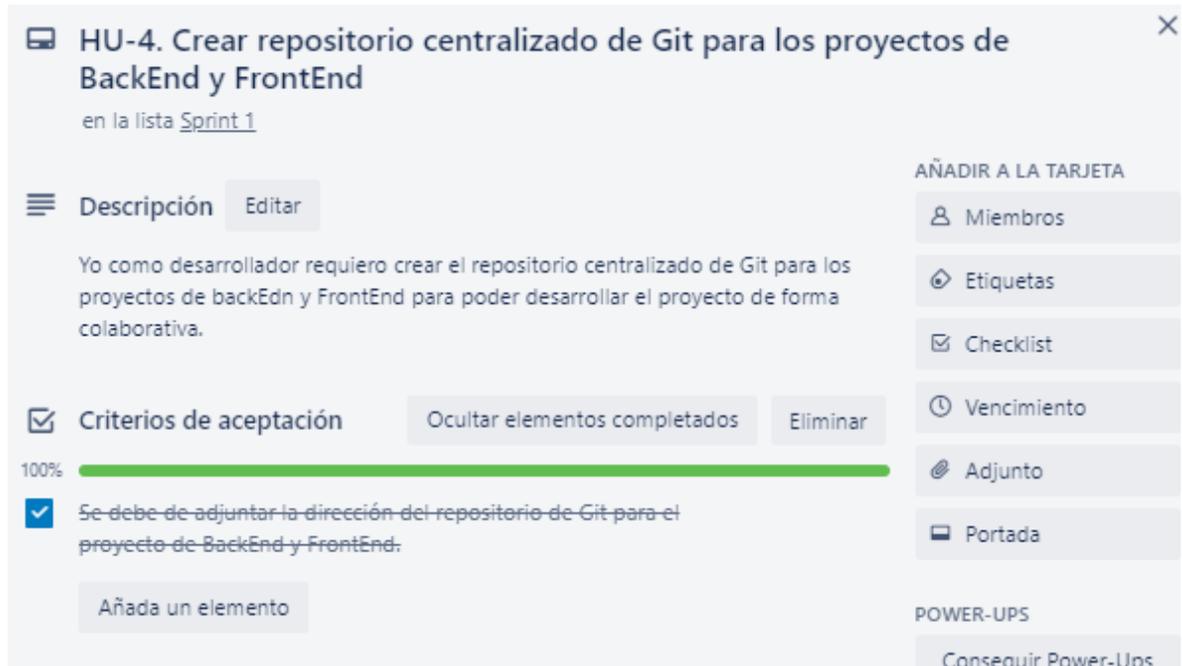


Figura 7.1.4

Autor: Auditoría propia, herramienta usada: Trello.

- HU 5

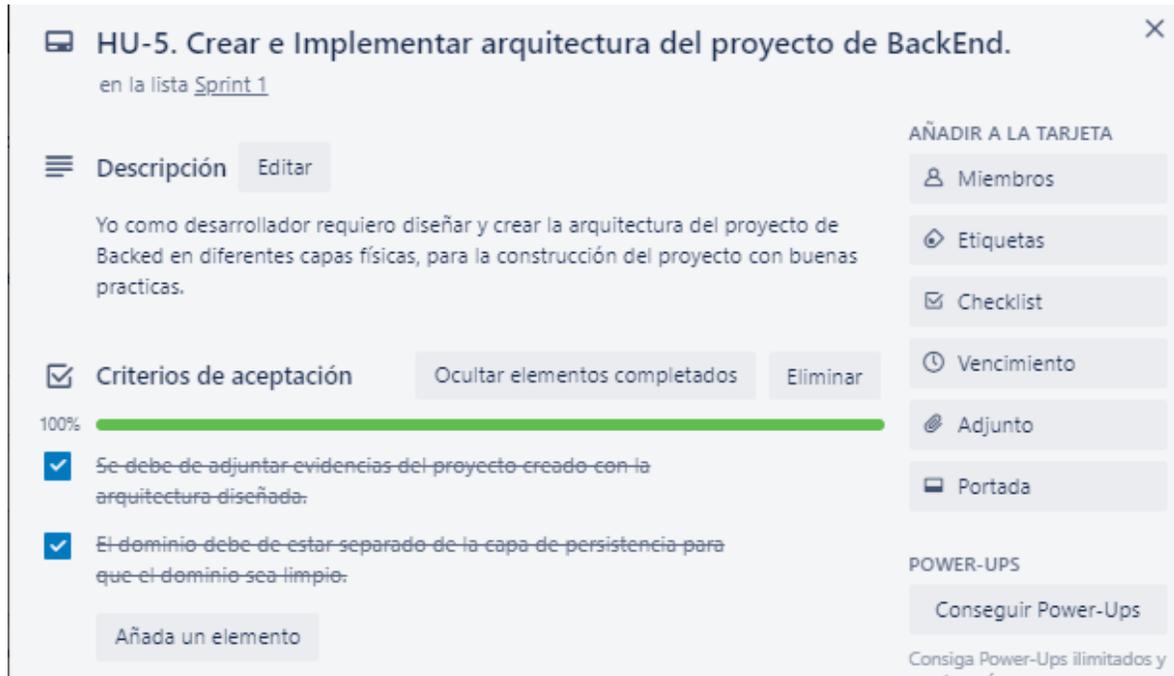


Figura 7.1.5

Autor: Auditoría propia, herramienta usada: Trello.

- HU 6

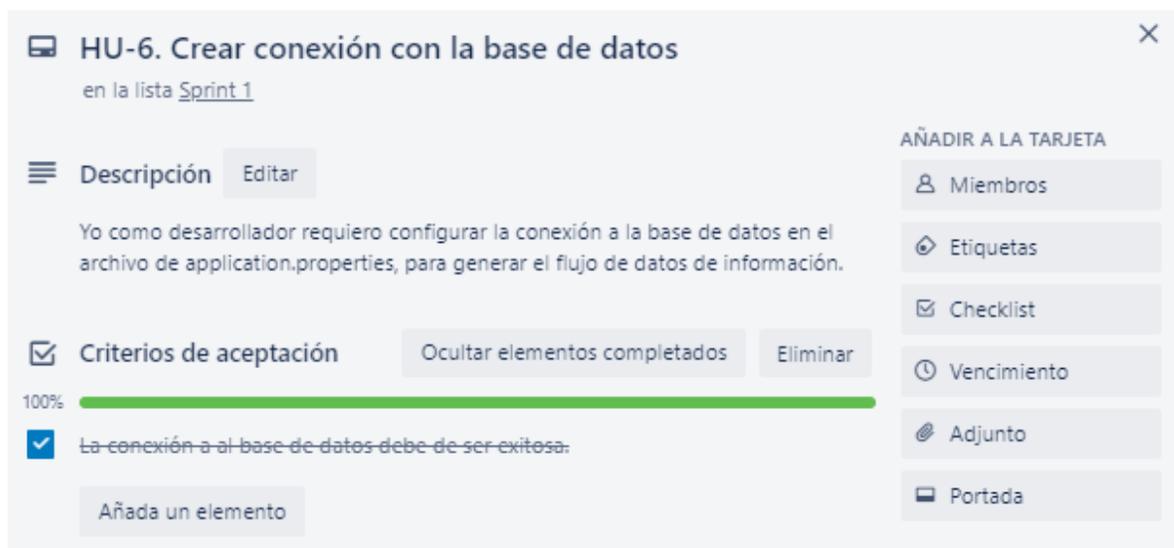


Figura 7.1.6

Autor: Auditoría propia, herramienta usada: Trello.

- HU 7

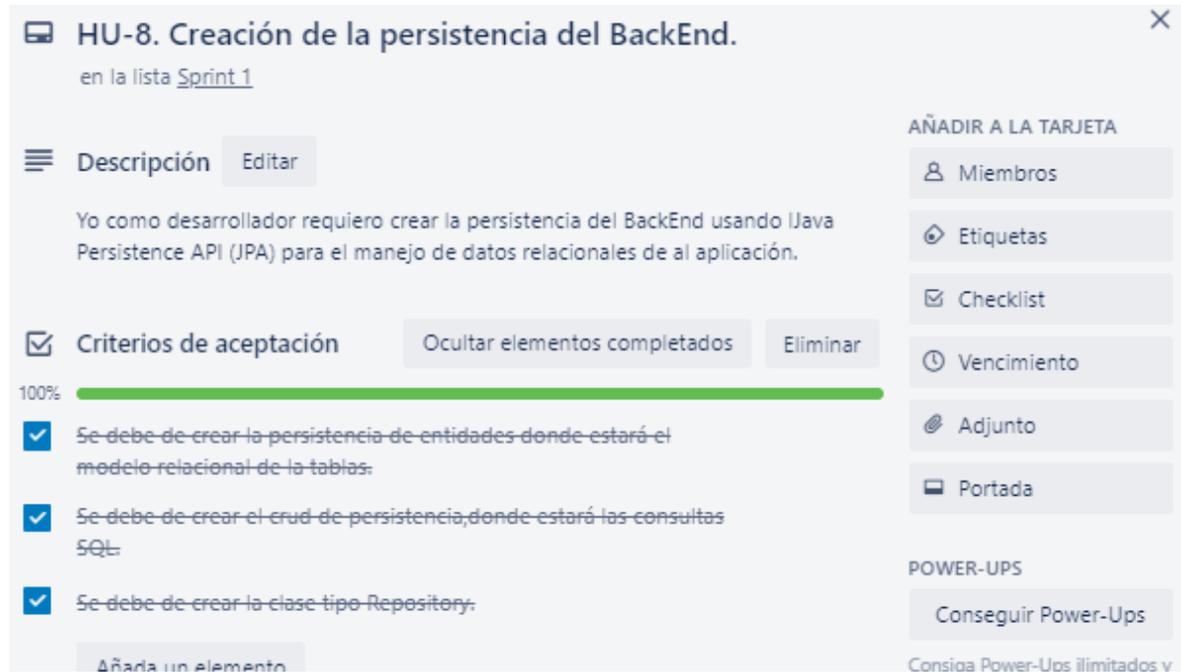
The screenshot shows a Trello card with the following content:

- Title:** HU-7. Creación de los dominios del BackEnd
- Location:** en la lista [Sprint 1](#)
- Description:** Yo como desarrollador requiero crear las clases de dominio necesarias para que la aplicación cumpla con las características que se requieren.
- Acceptance Criteria:**
  - El dominio debe de corresponder con lo que se va almacenar en la base de datos.
  - Los atributos de cada clase debe de coincidir con los nombres de los atributos y el tipo de dato de la base de datos.
- Progress:** 100% (indicated by a green bar)
- Buttons:** Descripción, Editar, Ocultar elementos completados, Eliminar, Añada un elemento
- Right Panel (Add to Card):** Miembros, Etiquetas, Checklist, Vencimiento, Adjunto, Portada
- Right Panel (Power-Ups):** Conseguir Power-Ups

Figura 7.1.7

Autor: Auditoría propia, herramienta usada: Trello.

- HU 8



**HU-8. Creación de la persistencia del BackEnd.**  
en la lista [Sprint 1](#)

**Descripción** Editar

Yo como desarrollador requiero crear la persistencia del BackEnd usando JJava Persistence API (JPA) para el manejo de datos relacionales de la aplicación.

**Criterios de aceptación** Ocultar elementos completados Eliminar

100%

- Se debe de crear la persistencia de entidades donde estará el modelo relacional de la tablas.
- Se debe de crear el crud de persistencia, donde estará las consultas SQL.
- Se debe de crear la clase tipo Repository.

Añade un elemento

**AÑADIR A LA TARJETA**

- Miembros
- Etiquetas
- Checklist
- Vencimiento
- Adjunto
- Portada

**POWER-UPS**

Conseguir Power-Ups

Consiga Power-Ups ilimitados v

Figura 7.1.8

Autor: Auditoría propia, herramienta usada: Trello.

- HU 9

**HU-9. Implementación de Dozer Api en la persistencia.**  
en la lista [Sprint 1](#)

**Descripción** Editar

Yo como desarrollador requiero implementar Dozer en la capa de la persistencia para definir un fichero de Mapeo que se encargue del tipo de conversión de los datos.

**Adjuntos**

**image.png** ↗  
Añadido: 2 de jun. a las 22:35 - [Comentario](#) - [Eliminar](#) - [Editar](#)  
[Crear portada](#)

Añada un adjunto

**Criterios de aceptación** Ocultar elementos completados Eliminar

100%

- ✓ Se debe de crear en la persistencia dos clases para el casteo de los datos, una para el repositorio base y otra con la configuración del Dozer.
- ✓ Se debe de evidenciar en el archivo de recursos de la persistencia los archivos tipo .xml de cada una de las clases de las entidades de dominio.

Añada un elemento

**AÑADIR A LA TARJETA**

- Miembros
- Etiquetas
- Checklist
- Vencimiento
- Adjunto
- Portada

**POWER-UPS**

Conseguir Power-Ups

Consiga Power-Ups ilimitados y mucho más.

[Actualizar el equipo](#)

**ACCIONES**

- Mover
- Copiar
- Convertir en plant...
- Seguir

Figura 7.1.9

Autor: Auditoría propia, herramienta usada: Trello.

- HU 10

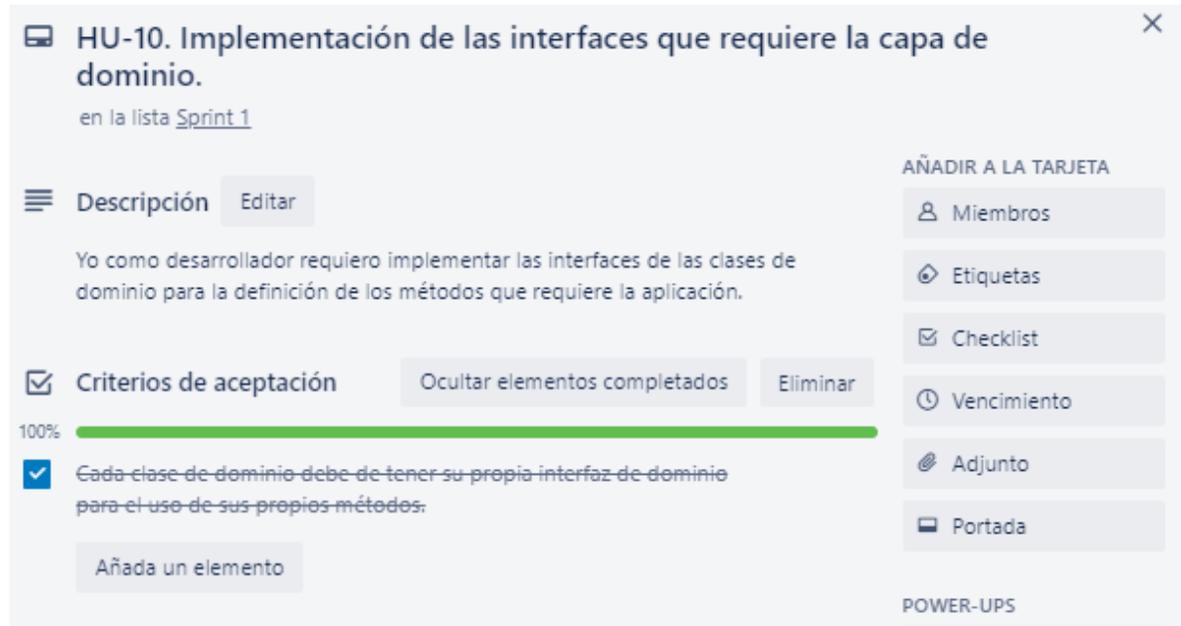


Figura 7.1.10

Autor: Auditoría propia, herramienta usada: Trello.

## Sprint 2

- HU 11

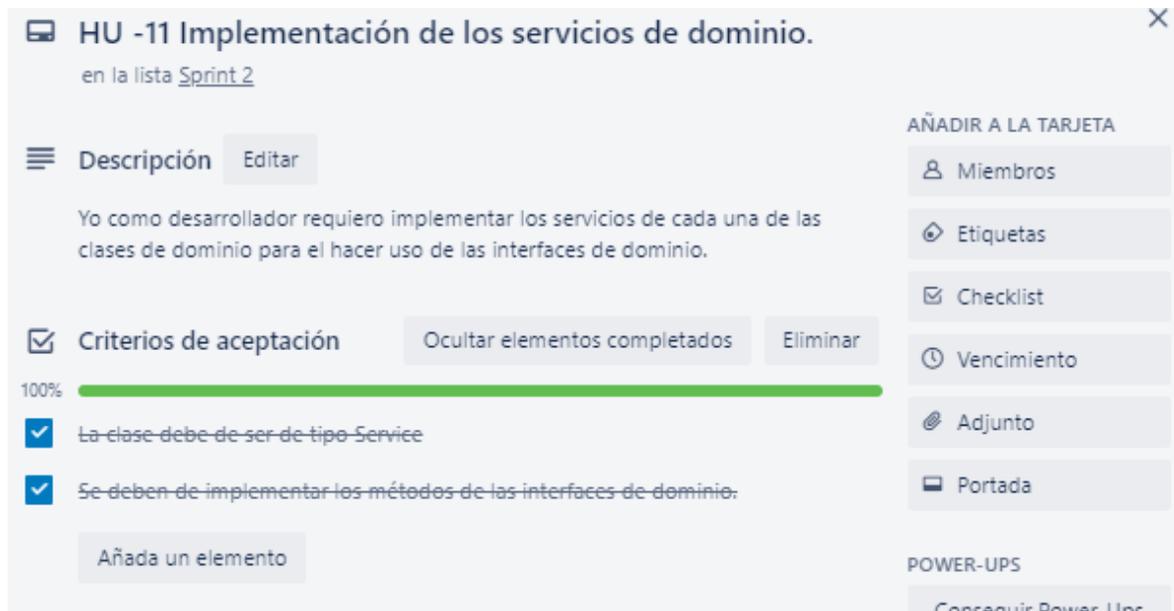


Figura 7.1.11

Autor: Auditoría propia, herramienta usada: Trello.

- HU 12

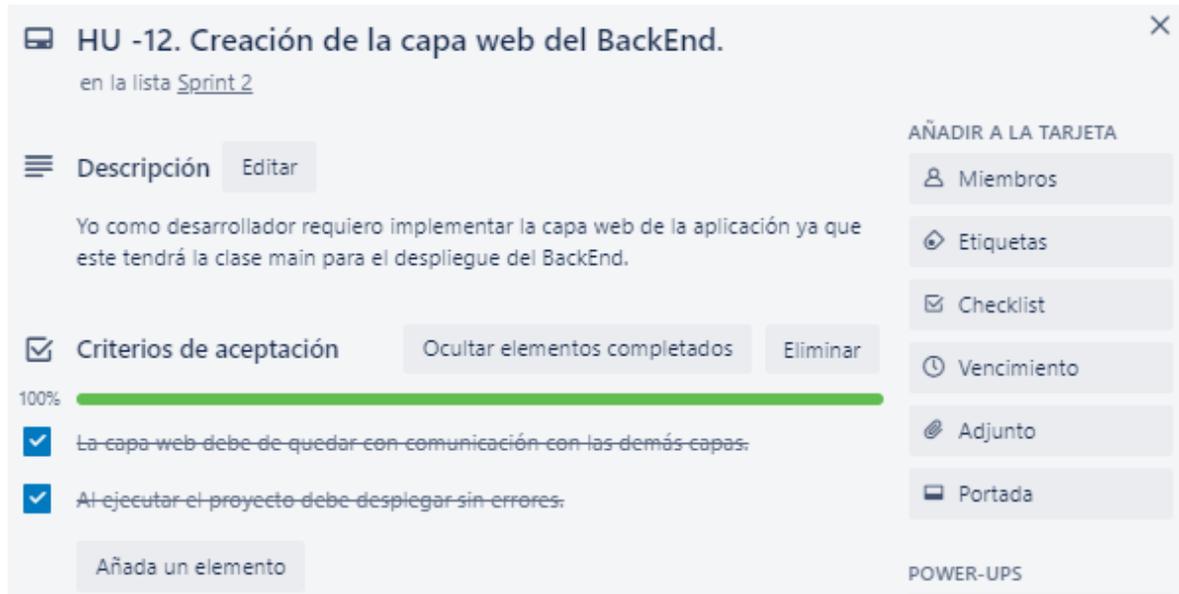


Figura 7.1.12

Autor: Auditoría propia, herramienta usada: Trello.

- HU 13

**HU -13. Implementación del controlador de los servicios web**  
en la lista [Sprint 2](#)

**Descripción** Editar

Yo como desarrollador requiero implementar el controlador de los servicios web de cada uno de los modelos que requiere la aplicación para que el FrontEnd los consuma.

Nota: Solo se debe de generar la clase RestController ,aun no se realizarán la construcción de los servicios.

**Criterios de aceptación** Ocultar elementos completados Eliminar

100%

- Se debe de especificar el RestController
- Se debe de generar la ruta del EndPoint para la comunicación que requiere el proyecto del FrontEnd.

Añada un elemento

**AÑADIR A LA TARJETA**

- Miembros
- Etiquetas
- Checklist
- Vencimiento
- Adjunto
- Portada

**POWER-UPS**

Conseguir Power-Ups

Consiga Power-Ups ilimitados y mucho más.

Figura 7.1.13

Autor: Auditoría propia, herramienta usada: Trello.

- HU 14

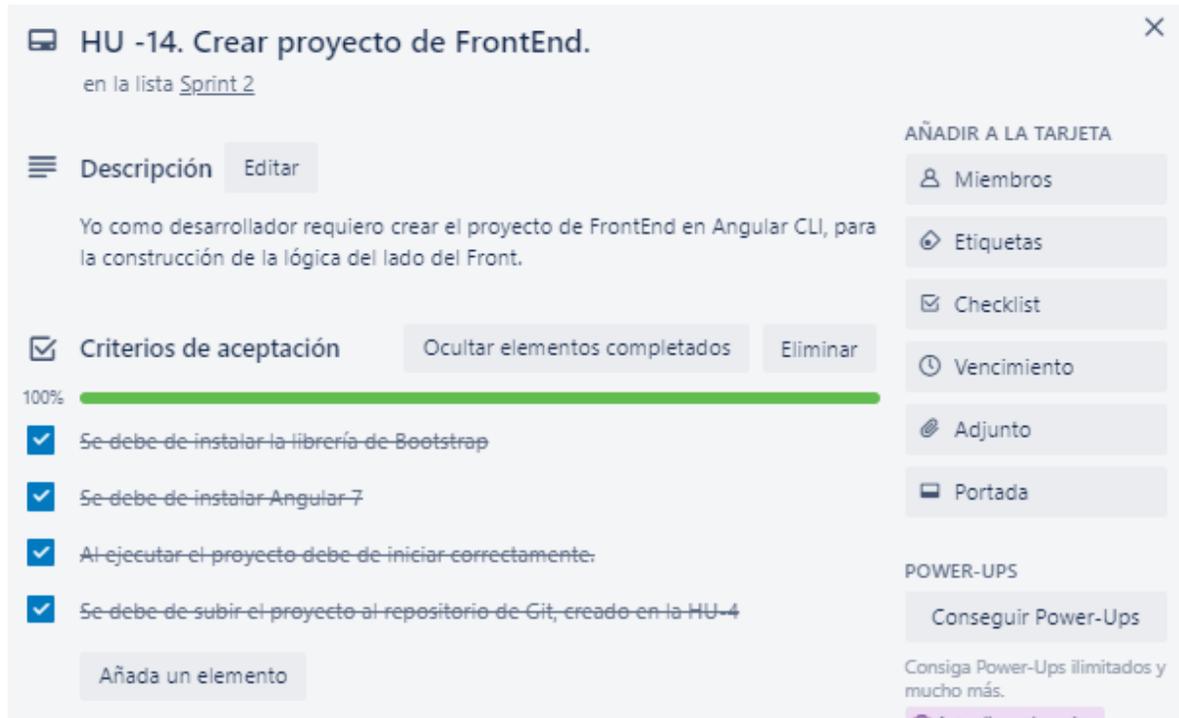
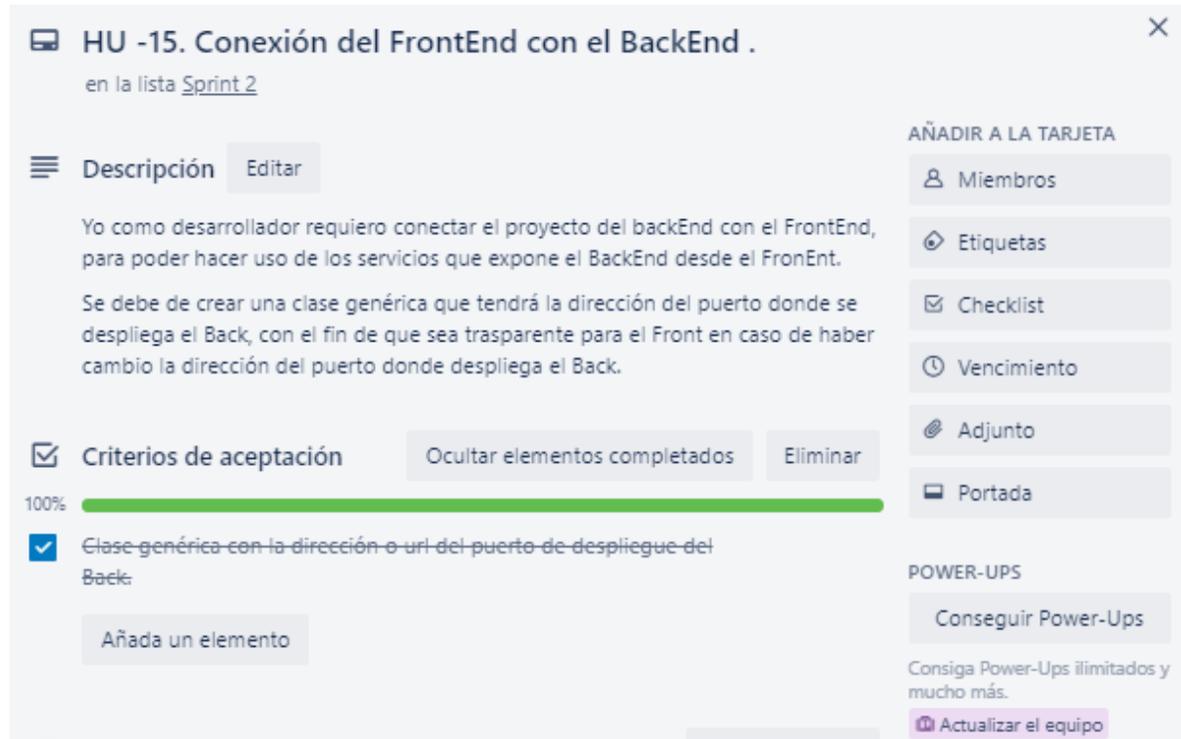


Figura 7.1.14

Autor: Auditoría propia, herramienta usada: Trello.

- HU 15



The image shows a Trello card interface. At the top, the card title is "HU -15. Conexión del FrontEnd con el BackEnd ." with a close button (X) in the top right corner. Below the title, it says "en la lista [Sprint 2](#)".

On the left side, there is a menu icon and the word "Descripción" with an "Editar" button next to it. The description text reads: "Yo como desarrollador requiero conectar el proyecto del backEnd con el FrontEnd, para poder hacer uso de los servicios que expone el BackEnd desde el FronEnt. Se debe de crear una clase genérica que tendrá la dirección del puerto donde se despliega el Back, con el fin de que sea transparente para el Front en caso de haber cambio la dirección del puerto donde despliega el Back."

Below the description, there is a section for "Criterios de aceptación" with a checkmark icon, an "Ocultar elementos completados" button, and an "Eliminar" button. A progress bar shows "100%" completion. A single checklist item is visible: "Clase genérica con la dirección o url del puerto de despliegue del Back." with a blue checkmark. Below this item is a button that says "Añada un elemento".

On the right side, there is a sidebar with the heading "AÑADIR A LA TARJETA". It contains several buttons: "Miembros", "Etiquetas", "Checklist", "Vencimiento", "Adjunto", and "Portada". Below this sidebar is a "POWER-UPS" section with a button "Conseguir Power-Ups" and a promotional message: "Consiga Power-Ups ilimitados y mucho más." followed by a button "Actualizar el equipo".

Figura 7.1.15

Autor: Auditoría propia, herramienta usada: Trello.

- HU 16

**HU-16. Crear vista del menú principal**  
en la lista [Sprint 2](#)

**Descripción** Editar

Yo como Administrador de la empresa requiero que la aplicación tenga un menú principal que contenga lo siguiente:

- Módulo de producción.
- Módulo de talento humano.
- Módulo de compras e inventario.

Para clasificar la aplicaciones por funciones y módulos.

**Criterios de aceptación** Ocultar elementos completados Eliminar

100%

- Cuando se de clic en el módulo de producción, este debe de direccionar a la vista de producción.
- Cuando se de clic en el módulo de Talento humano, este debe de direccionar a la vista de Talento humano.
- Cuando se de clic en el módulo de Compras e Inventario, este debe de direccionar a la vista de Compras e Inventario.
- El menú de estos módulos debe de ser imágenes.

**AÑADIR A LA TARJETA**

- Miembros
- Etiquetas
- Checklist
- Vencimiento
- Adjunto
- Portada

**POWER-UPS**

Conseguir Power-Ups

Consiga Power-Ups ilimitados y mucho más.

Actualizar el equipo

**ACCIONES**

- Mover

Figura 7.1.16

Autor: Auditoría propia, herramienta usada: Trello.

- HU 17

**HU -17. Crear vista del módulo de producción.**  
en la lista [Sprint 2](#)

**Descripción** Editar

Yo como administrador del sistema requiero que la aplicación tenga un módulo de producción que permita ingresar a:

- La producción de Mojes
- Los productos terminados.
- Las fórmulas de la panadería.

Para poder ingresar a cada área específica de producción.

**Criterios de aceptación** Ocultar elementos completados Eliminar

100%

- ✓ Cuando se de clic en la opción Moje, este debe de mostrar dos opciones a seleccionar: 1. si deseo ingresar a fabricar un moje → 2. Si deseo ingresar al inventario de mojes.
- ✓ Cuando se de clic en la opción Producto Terminado, este debe de mostrar dos opciones a seleccionar: 1. si deseo ingresar a ver los productos terminados → 2. Si deseo ingresar a crear un producto terminado.
- ✓ Cuando se de clic en la opción de Formulas, este debe de mostrar tres opciones a seleccionar: 1. si deseo ingresar a ver las formulas → 2. Si deseo ingresar a crear una nueva formula. → 3. Si deseo ingresar a ver los producto que tengo en el sistema.
- ✓ Cada item de opción debe de ser una imagen.

Añada un elemento

**AÑADIR A LA TARJETA**

- Miembros
- Etiquetas
- Checklist
- Vencimiento
- Adjunto
- Portada

**POWER-UPS**

Conseguir Power-Ups

Consiga Power-Ups ilimitados y mucho más.

- Actualizar el equipo

**ACCIONES**

- Mover
- Copiar
- Convertir en plant...
- Seguir

Figura 7.1.17

Autor: Auditoría propia, herramienta usada: Trello.

- HU 18

**HU - 18. Opciones del módulo de producción.**  
en la lista [Sprint 2](#)

**Descripción** Editar

Yo como administrador del sistema requiero que las opciones del moduló de producción me direccionen a su respectiva vista para poder acceder a ellas.  
Las opciones fueron creadas anteriormente en la HU-17

**Criterios de aceptación** Ocultar elementos completados Eliminar

100%

- ✓ Al dar clic en la opción fabricar moje esta debe de direccionar a la vista de fabricar moje;
- ✓ Al dar clic en la opción de inventario moje, esta debe de direccionar a la vista del inventario de mojes;
- ✓ Al dar clic en la opción de ver producto final, esta debe de direccionar a la vista de productos terminados;
- ✓ Al dar clic en la opción de crear producto final, esta debe de direccionar a la vista de fabricar producto final;
- ✓ Al dar clic en la opción ver formulas, esta debe de direccionar a la vista de ver fórmulas;
- ✓ Al dar clic en la opción crear formula, esta debe de direccionar a la vista de crear formula
- ✓ Al dar clic en la opción ver productos, esta debe de direccionar a la vista de productos;

Añada un elemento

**AÑADIR A LA TARJETA**

- Miembros
- Etiquetas
- Checklist
- Vencimiento
- Adjunto
- Portada

**POWER-UPS**

Conseguir Power-Ups

Consiga Power-Ups ilimitados y mucho más.

Actualizar el equipo

**ACCIONES**

- Mover
- Copiar
- Convertir en plant...
- Seguir

Figura 7.1.18

Autor: Auditoría propia, herramienta usada: Trello.

- HU 19

**HU - 19. Crear vista del módulo de talento humano.**  
en la lista [Sprint 2](#)

**Descripción** Editar

Yo como administrador del sistema requiero que la aplicación tenga un módulo de talento humano que permita ingresar a :

- Personas
- Empleados
- Utilidades

Para poder ingresar a cada área específica de talento humano.

**Criterios de aceptación** Ocultar elementos completados Eliminar

100%

- ✓ Cuando se de clic en la opción Persona, este debe de mostrar dos opciones a seleccionar: 1. si deseo ingresar a ver las personas → 2. Si deseo ingresar a crear una nueva persona.
- ✓ Cuando se de clic en la opción Empleados, este debe de mostrar dos opciones a seleccionar: 1. si deseo ingresar a ver los Empleados → 2. Si deseo ingresar a crear un nuevo Empleado.
- ✓ Cuando se de clic en la opción de Utilidades, este debe de mostrar cuatro opciones a seleccionar: 1. si deseo ingresar a ver los Tipos de personas → 2. Si deseo ingresar a ver los tipo de documentos. → 3. Si deseo ingresar a ver los tipos de teléfonos. → 4. Si deseo ingresar a ver lo roles.
- ✓ Cada item de opción debe de ser una imagen.

Añada un elemento

**AÑADIR A LA TARJETA**

- Miembros
- Etiquetas
- Checklist
- Vencimiento
- Adjunto
- Portada

**POWER-UPS**

Conseguir Power-Ups

Consiga Power-Ups ilimitados y mucho más.

Actualizar el equipo

**ACCIONES**

- Mover
- Copiar
- Convertir en plant...
- Seguir

Figura 7.1.19

Autor: Auditoría propia, herramienta usada: Trello.

- HU 20

**HU - 20. Opciones del módulo de talento humano.**  
en la lista [Sprint 2](#)

**Descripción** Editar

Yo como administrador del sistema requiero que las opciones del moduló de producción me direccionen a su respectiva vista para poder acceder a ellas.

Las opciones fueron creadas anteriormente en la HU-19

**Criterios de aceptación** Ocultar elementos completados Eliminar

100%

- ✓ Al dar clic en la opción Ver Personas esta debe de direccionar a la vista de Ver Personas;
- ✓ Al dar clic en la opción de Crear Persona, esta debe de direccionar a la vista de Crear Persona
- ✓ Al dar clic en la opción de Empleados, esta debe de direccionar a la vista de Ver Empleados;
- ✓ Al dar clic en la opción de Crear Empleado, esta debe de direccionar a la vista Crear un nuevo Empleado;
- ✓ Al dar clic en la opción Tipos de personas, esta debe de direccionar a la vista de Ver tipos de personas;
- ✓ Al dar clic en la opción Tipos de documentos, esta debe de direccionar a la vista de ver Tipos de documentos;
- ✓ Al dar clic en la opción Tipos de teléfonos, esta debe de direccionar a la vista Ver tipos de teléfonos;
- ✓ Al dar clic en la opción de roles, esta debe de direccionar a la vista de ver Roles;

Añada un elemento

**AÑADIR A LA TARJETA**

- Miembros
- Etiquetas
- Checklist
- Vencimiento
- Adjunto
- Portada

**POWER-UPS**

Conseguir Power-Ups

Consiga Power-Ups ilimitados y mucho más.

Actualizar el equipo

**ACCIONES**

- Mover
- Copiar
- Convertir en plant...
- Seguir
- Archivar

Figura 7.1.20

Autor: Auditoría propia, herramienta usada: Trello.

## Sprint 3

- HU 21

**HU - 21. Crear vista del módulo de Compras e Inventario.**  
en la lista [Sprint 3](#)

**Descripción** Editar

Yo como administrador del sistema requiero que la aplicación tenga un módulo de compras e inventario que permita ingresar a :

- Compras.
- Inventario de materias primas.
- Proveedores.

Para poder ingresar a cada área específica de compras e inventario.

**Criterios de aceptación** Ocultar elementos completados Eliminar

100%

- ✓ Cuando se de clic en la opción Compras, este debe de mostrar dos opciones a seleccionar: 1. Ingresar a ver las compra-2. Ingresar a crear una nueva compra.
- ✓ Cuando se de clic en la opción Inventario Materia Prima este debe de mostrar cuatro opciones a seleccionar: 1. Ingresar a ver el inventario de materias primas-2. Ingresar a crear una nueva materia prima.-3. Ingresar a ver el inventario bajo.-4. Ingresar a ver las unidades de medidas.
- ✓ Cuando se de clic en la opción de Proveedores, este debe de mostrar dos opciones a seleccionar: 1. Ingresar a ver los los proveedores-2. Ingresar a crear un nuevo proveedor.
- ✓ Cada item de opción debe de ser una imagen.

Añada un elemento

**AÑADIR A LA TARJETA**

- Miembros
- Etiquetas
- Checklist
- Vencimiento
- Adjunto
- Portada

**POWER-UPS**

Conseguir Power-Ups

Consiga Power-Ups ilimitados y mucho más.

Actualizar el equipo

**ACCIONES**

- Mover
- Copiar
- Convertir en plant...
- Seguir
- Archivar

Figura 7.1.21

Autor: Auditoría propia, herramienta usada: Trello.

- HU 22

**HU - 22. Opciones del módulo de compras e inventario.**  
en la lista [Sprint 3](#)

**Descripción** Editar

Yo como administrador del sistema requiero que las opciones del módulo de compras e inventario me direccionen a su respectiva vista para poder acceder a ellas.

Las opciones fueron creadas anteriormente en la HU-21

**Criterios de aceptación** Ocultar elementos completados Eliminar

100%

- Al dar clic en la opción Ver Compras, esta debe de direccionar a la vista de Ver Compras;
- Al dar clic en la opción de Crear Compra, esta debe de direccionar a la vista de Crear Compra;
- Al dar clic en la opción de Inventario Materia Prima, esta debe de direccionar a la vista de Ver Inventario de MP;
- Al dar clic en la opción de Crear Materia Prima, esta debe de direccionar a la vista de Crear Materia Prima;
- Al dar clic en la opción Inventario Bajo, esta debe de direccionar a la vista de Ver Inventario Bajo;
- Al dar clic en la opción Unidades de Medida, esta debe de direccionar a la vista de ver Unidades de Medida;
- Al dar clic en la opción Ver Proveedores, esta debe de direccionar a la vista de Ver Proveedores;
- Al dar clic en la opción de Crear Proveedor, esta debe de direccionar a la vista Crear Proveedor;

Añada un elemento

**AÑADIR A LA TARJETA**

- Miembros
- Etiquetas
- Checklist
- Vencimiento
- Adjunto
- Portada

**POWER-UPS**

Conseguir Power-Ups

Consiga Power-Ups ilimitados y mucho más.

Actualizar el equipo

**ACCIONES**

- Mover
- Copiar
- Convertir en plant...
- Seguir
- Archivar
- Compartir

Figura 7.1.22

Autor: Auditoría propia, herramienta usada: Trello.

- HU 23

**HU - 23. Ingresar una nueva compra.**  
en la lista [Sprint 3](#)

**Descripción** Editar

Yo como administrador del sistema requiero que el sistema me permita ingresar una nueva compra, para poder manejar un inventario de los insumos o materia prima de mi sede.

Para poder ingresar la compra se necesita tener un proveedor de la materia prima a comprar, el sistema debe de mostrar los proveedores que hay disponibles en la sede y permitir seleccionar uno de ellos.

**Criterios de aceptación** Ocultar elementos completados Eliminar

100%

- El sistema me debe de mostrar los proveedores disponibles y permitirme elegir uno de ellos.
- El sistema debe de pedir los siguientes campos a ingresar: Proveedor, Fecha de compra, Número de factura, Descripción y Valor total de la compra.
- Todos los campos deben ser obligatorios.
- El campo de fecha debe de tener formato dd/mm/aaaa
- El campo de número de factura solo aceptará facturas que tengan por lo menos 6 caracteres. Se aceptan números y letras.
- El campo de descripción solo aceptará una descripción que tengan por lo menos 6 caracteres. Se aceptan números, letras y símbolos.
- El campo de valor de compra sólo aceptará números.
- Botón para crear la compra.
- Botón para cancelar el ingreso de la compra.

Añada un elemento

**AÑADIR A LA TARJETA**

- Miembros
- Etiquetas
- Checklist
- Vencimiento
- Adjunto
- Portada

**POWER-UPS**

Conseguir Power-Ups

Consiga Power-Ups ilimitados y mucho más.

Actualizar el equipo

**ACCIONES**

- Mover
- Copiar
- Convertir en plant...
- Seguir
- Archivar
- Compartir

Figura 7.1.23

Autor: Auditoría propia, herramienta usada: Trello.

- HU 24

**HU - 24. Crear vista para ver compras.**  
en la lista [Sprint 3](#)

**Descripción** Editar

Yo como administrador del sistema requiero que la aplicación permita observar las compras que se han realizado en la sede para poder conocer que se ha comprado..

**Criterios de aceptación** Ocultar elementos completados Eliminar

100%

- ✓ Se deben de observar las compras de esa sede.
- ✓ Se debe de observar por compra la siguiente información: Proveedor, Fecha de compra, Número de factura, Descripción y Precio total.
- ✓ La información de la compra observada tiene que coincidir con la compra que se ingreso.
- ✓ Filtro para buscar una compra por el nombre del proveedor.
- ✓ Filtro para buscar una compra por fecha de compra. El formato de búsqueda tiene que ser dd/mm/aaaa.
- ✓ Botón que permita crear una nueva compra.
- ✓ Botón para volver al módulo de compras e inventario.

Añada un elemento

**AÑADIR A LA TARJETA**

- Miembros
- Etiquetas
- Checklist
- Vencimiento
- Adjunto
- Portada

**POWER-UPS**

Conseguir Power-Ups

Consiga Power-Ups ilimitados y mucho más.

Actualizar el equipo

**ACCIONES**

- Mover
- Copiar

Figura 7.1.24

Autor: Auditoría propia, herramienta usada: Trello.

- HU 25

**HU - 25. Implementar opciones al seleccionar una compra.**  
en la lista [Sprint 3](#)

**Descripción** Editar

Yo como administrador del sistema requiero que la aplicación permita seleccionar una compra para ver el detalle de la compra, o editar la compra o eliminar dicha compra.

Nota: El botón de eliminar estará activó siempre y cuando no se halla generado un detalle de la compra. Si se ingresó un detalle de compra el botón de eliminar desaparecerá.

**Criterios de aceptación** Ocultar elementos completados Eliminar

100%

- ✓ Botón "Detalles compra" para ingresar a observar el detalle de la compra seleccionada.
- ✓ Botón "Editar" para actualizar la información de la compra seleccionada.
- ✓ Botón "Eliminar" para borrar la compra seleccionada.
- ✓ Al dar clic en el botón de Detalles compra, este debe de direccionar a la vista de Detalles compra.
- ✓ Al dar clic en el botón de Editar, este debe de direccionar a la vista de Editar compra.
- ✓ Al dar clic en el botón Eliminar, este debe de mostrar una ventana emergente de confirmación. Si se da aceptar el sistema debe de eliminar la compra de la base de datos.
- ✓ Si hay un detalle de compra el botón de eliminar no se debe mostrar.

Añada un elemento

**AÑADIR A LA TARJETA**

- Miembros
- Etiquetas
- Checklist
- Vencimiento
- Adjunto
- Portada

**POWER-UPS**

Conseguir Power-Ups

Consiga Power-Ups ilimitados y mucho más.

- Actualizar el equipo

**ACCIONES**

- Mover
- Copiar
- Convertir en plant...
- Seguir
- Archivar

Figura 7.1.25

Autor: Auditoría propia, herramienta usada: Trello.

- HU 26

**HU - 26 .Editar o actualizar información de una compra.**  
en la lista [Sprint 3](#)

**Descripción** Editar

Yo como administrador del sistema requiero que la aplicación permita actualizar la información de una compra, para corregir información mal ingresada.

**Criterios de aceptación** Ocultar elementos completados Eliminar

100%

- ✓ Se deben debe de mostrar la información de la compra con todos sus campos.
- ✓ El sistema me debe de mostrar los proveedores disponibles y permitirme actualizar el actual.
- ✓ El campo de fecha debe de tener formato dd/mm/aaaa en caso de ser actualizada.
- ✓ Si se actualiza el campo de número de factura, este solo aceptará facturas que tengan por lo menos 6 caracteres. Se aceptan números y letras.
- ✓ Si se actualiza el campo de descripción, este solo aceptará una descripción que tengan por lo menos 6 caracteres. Se aceptan números, letras y símbolos.
- ✓ Si se actualiza el campo de valor de compra, este sólo aceptará números.
- ✓ No se aceptarán campos en blanco (Nulos)
- ✓ Botón para Actualizar la información de la compra.
- ✓ Botón para cancelar la transacción.
- ✓ Al dar clic en el botón de cancelar este debe de direccionar a la vista de ver Compras.
- ✓ La información debe de quedar actualizada correctamente.

Añada un elemento

**AÑADIR A LA TARJETA**

- Miembros
- Etiquetas
- Checklist
- Vencimiento
- Adjunto
- Portada

**POWER-UPS**

Conseguir Power-Ups

Consiga Power-Ups ilimitados y mucho más.

Actualizar el equipo

**ACCIONES**

- Mover
- Copiar
- Convertir en plant...
- Seguir
- Archivar
- Compartir

Figura 7.1.26

Autor: Auditoría propia, herramienta usada: Trello.

- HU 27

**HU - 27. ingresar un detalle de una compra.**  
en la lista [Sprint 3](#)

**Descripción** [Editar](#)

Yo como administrador del sistema necesito que la aplicación permita ingresar el detalle de una compra realizada, para poder registrar lo que se compró en una factura de compra.

El sistema debe de permitir seleccionar un insumo para añadirlo a la compra.

**Criterios de aceptación** [Ocultar elementos completados](#) [Eliminar](#)

100%

- Lista de los insumos que tiene el sistema para agregarlos a una compra detalle.
- Al seleccionar un insumo debe de permitir ingresar los siguientes datos: Cantidad en gramos, Valor de la compra de ese insumo, fecha de vencimiento del insumo, Observaciones de ese insumo comprado.
- El campo de cantidad solo aceptará ingresar números.
- El campo valor compra, solo permitirá ingresar números.
- El campo fecha de vencimiento debe de ser en formato dd/mm/aaaa.
- El campo de observaciones sólo permitirá ingresar una observación con al menos 4 caracteres.
- Todos los campos son obligatorios.
- Botón para crear o añadir ese insumo a la compra.
- Botón para cancelar el ingreso del insumo.
- Filtro para buscar el insumo que se necesita ingresar al detalle de la compra.

[Añada un elemento](#)

**AÑADIR A LA TARJETA**

- [Miembros](#)
- [Etiquetas](#)
- [Checklist](#)
- [Vencimiento](#)
- [Adjunto](#)
- [Portada](#)

**POWER-UPS**

[Conseguir Power-Ups](#)

Consiga Power-Ups ilimitados y mucho más.

[Actualizar el equipo](#)

**ACCIONES**

- [Mover](#)
- [Copiar](#)
- [Convertir en plant...](#)
- [Seguir](#)
- [Archivar](#)
- [Compartir](#)

Figura 7.1.27

Autor: Auditoría propia, herramienta usada: Trello.

- HU 28

**HU-28 Vista para ver los detalles de una compra.**  
en la lista [Sprint 3](#)

**Descripción** Editar

Yo como administrador del sistema requiero que la aplicación permita observar el detalle de una compra seleccionada, para saber que insumos se compraron y el valor que costo cada uno de ellos.

**Criterios de aceptación** Ocultar elementos completados Eliminar

100%

- Se deben de observar los insumos o materias primas comprados en esa factura de compra.
- Se debe de observar por insumo o materia prima comprada, la siguiente información: Materia prima, Cantidad, Precio, Fecha de vencimiento y Observación.
- La información de cada insumo o materia prima comprado debe de coincidir con el insumo que se ingreso al detalle.
- Botón "Crear detalle compra" que permita agregar otro insumo al detalle de la compra.
- Botón para volver a la vista de Compras.

Añada un elemento

**AÑADIR A LA TARJETA**

- Miembros
- Etiquetas
- Checklist
- Vencimiento
- Adjunto
- Portada

**POWER-UPS**

Conseguir Power-Ups

Consiga Power-Ups ilimitados y mucho más.

Actualizar el equipo

**ACCIONES**

- Mover
- Copiar

Figura 7.1.28

Autor: Auditoría propia, herramienta usada: Trello.

- HU 29

**HU- 29. Vista para ver el inventario de materias primas.**  
en la lista [Sprint 3](#)

**Descripción** Editar

Yo como administrador del sistema requiero que la aplicación permita observar las materia primas que hay en el inventario para tener conocimiento de los insumos que posee la panadería.

**Criterios de aceptación** Ocultar elementos completados Eliminar

100%

- ✓ Se debe de observar las materias primas que hay en el inventario.
- ✓ La información de cada materia prima debe de coincidir con lo que se ingresó.
- ✓ Por cada materia prima se debe de mostrar la siguiente información: Nombre, Unidad, Cantidad, Mínimo de inventario, Valor unidad y Maneja inventario.
- ✓ Filtro para buscar por el nombre de una materia prima.
- ✓ Botón para ingresar una nueva materia prima.
- ✓ Botón para regresar al módulo de compras e inventario.

Añada un elemento

**AÑADIR A LA TARJETA**

- Miembros
- Etiquetas
- Checklist
- Vencimiento
- Adjunto
- Portada

**POWER-UPS**

Conseguir Power-Ups

Consiga Power-Ups ilimitados y mucho más.

Actualizar el equipo

**ACCIONES**

- Mover
- Copiar

Figura 7.1.29

Autor: Auditoría propia, herramienta usada: Trello.

- HU 30

**HU - 30. Implementar opciones al seleccionar una materia prima del inventario.**  
en la lista [Sprint 3](#)

**Descripción** Editar

Yo como administrador del sistema requiero que la aplicación permita seleccionar una materia prima del inventario para poder editarla o eliminarla.

Nota: El botón de eliminar estará activó siempre y cuando no se halla usado en un detalle de compra. Si se usó en un detalle de compra el botón de eliminar desaparecerá.

**Criterios de aceptación** Ocultar elementos completados Eliminar

100%

- ✓ Al dar clic en una materia prima se debe de mostrar dos botones: Editar y Eliminar.
- ✓ Botón "Editar" para actualizar la información de una materia prima.
- ✓ Al dar clic en el botón de Editar, este debe de direccionar a la vista de Editar Materia prima.
- ✓ Botón "Eliminar" para borrar la materia prima seleccionada.
- ✓ Al dar clic en el botón Eliminar, este debe de mostrar una ventana emergente de confirmación. Si se da aceptar el sistema debe de eliminar la materia prima del sistema.
- ✓ Si se uso la materia prima en un detalle de compra el botón de eliminar no se debe mostrar.

Añada un elemento

**AÑADIR A LA TARJETA**

- Miembros
- Etiquetas
- Checklist
- Vencimiento
- Adjunto
- Portada

**POWER-UPS**

Conseguir Power-Ups

Consiga Power-Ups ilimitados y mucho más.

Actualizar el equipo

**ACCIONES**

- Mover
- Copiar
- Convertir en plant...
- Seguir ✓

Figura 7.1.30

Autor: Auditoría propia, herramienta usada: Trello.

## Sprint 4

- HU 31

**HU - 31. Ingresar una materia prima al inventario.**  
en la lista [Sprint 4](#)

**Descripción** Editar

Yo como administrador del sistema requiero que la aplicación permita ingresar una materia prima para registrarla en el inventario de insumos de la panadería.

**Criterios de aceptación** Ocultar elementos completados Eliminar

100%

- ✓ La aplicación debe permitir ingresar la siguiente información: Nombre de la materia prima, Unidad de medida, Cantidad de materia prima, Cantidad mínima permitida para el inventario, Valor unidad y manejo inventario.
- ✓ Todos los campos son obligatorios.
- ✓ El campo "Nombre" solo aceptará un nombre con al menos 4 caracteres.
- ✓ El sistema debe de mostrar las unidades de medida disponibles y permitir elegir una de ellas;
- ✓ El campo "Cantidad", solo aceptará números y decimales.
- ✓ El campo "Mínimo en inventario", solo aceptará números y decimales.
- ✓ El campo "Valor unidad", solo aceptará números y decimales.
- ✓ El campo "Manejo inventario" debe de permitir elegir un SI o un No.
- ✓ Botón para crear o añadir la materia prima al inventario.
- ✓ Botón para cancelar el ingreso de la materia prima.

Añada un elemento

**AÑADIR A LA TARJETA**

- Miembros
- Etiquetas
- Checklist
- Vencimiento
- Adjunto
- Portada

**POWER-UPS**

Conseguir Power-Ups

Consiga Power-Ups ilimitados y mucho más.

Actualizar el equipo

**ACCIONES**

- Mover
- Copiar
- Convertir en plant...
- Seguir
- Archivar
- Compartir

Figura 7.1.31

Autor: Auditoría propia, herramienta usada: Trello.

- HU 32

**HU - 32. Editar o actualizar la información de una materia prima.**  
en la lista [Sprint 4](#)

**Descripción** Editar

Yo como administrador del sistema requiero que la aplicación permita actualizar la información de una materia prima ya registrada en el sistema., para corregir información mal ingresada.

**Criterios de aceptación** Ocultar elementos completados Eliminar

100%

- ✓ Se deben debe de mostrar la información de la materia prima con todos sus campos.
- ✓ El sistema me debe de mostrar las unidades de medida disponibles y permitirme actualizar la unidad de medida actual.
- ✓ Si se actualiza el nombre, este solo aceptará como mínimo un nombre de 4 caracteres.
- ✓ Si se actualiza el campo de cantidad, este solo aceptará números y decimales.
- ✓ Si se actualiza el campo de cantidad mínima de inventario , este solo aceptará números y decimales.
- ✓ Si se actualiza el campo de valor unidad, este sólo aceptará números y decimales.
- ✓ El campo maneja inventario debe permitir seleccionar "SI" o "NO" en caso de querer actualizarlo.
- ✓ No se aceptarán campos en blanco(Nulos)
- ✓ Botón para cancelar la transacción.
- ✓ Al dar clic en el botón de cancelar este debe de direccionar a la vista de ver Inventario materia prima.
- ✓ Botón para actualizar la información.
- ✓ La materia prima debe de quedar con los datos actualizados correctamente.

Añada un elemento

**AÑADIR A LA TARJETA**

- Miembros
- Etiquetas
- Checklist
- Vencimiento
- Adjunto
- Portada

**POWER-UPS**

Conseguir Power-Ups

Consiga Power-Ups ilimitados y mucho más.

Actualizar el equipo

**ACCIONES**

- Mover
- Copiar
- Convertir en plant...
- Seguir
- Archivar
- Compartir

Figura 7.1.32

Autor: Auditoría propia, herramienta usada: Trello.

- HU 33

**HU - 33. Ingresar una unidad de medida.**  
en la lista [Sprint 4](#)

**Descripción** Editar

Yo como administrador del sistema requiero que la aplicación permita ingresar unidades de medida, para tener control de la media en que se trabaja la cantidad de los insumos del inventario.

**Criterios de aceptación** Ocultar elementos completados Eliminar

100%

- Para ingresar la información se debe de tener el siguiente campo: Descripción.
- El campo descripción, solo aceptará una descripción mayor a cuatro caracteres,
- El campo descripción es obligatorio.
- Botón para crear o guardar a unidad de medida.
- Botón para cancelar el proceso.
- Al dar clic en el botón cancelar este debe de direccionar a la vista de unidades de medida.

Añada un elemento

**AÑADIR A LA TARJETA**

- Miembros
- Etiquetas
- Checklist
- Vencimiento
- Adjunto
- Portada

**POWER-UPS**

Conseguir Power-Ups

Consiga Power-Ups ilimitados y mucho más.

Actualizar el equipo

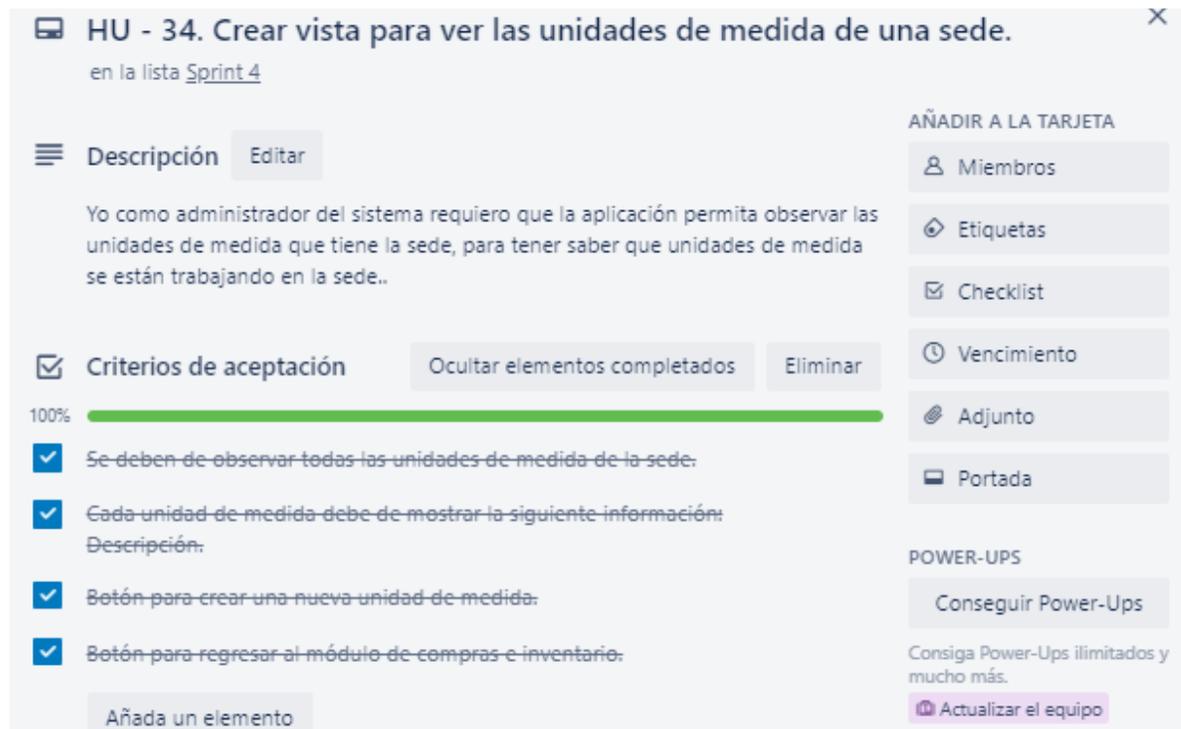
**ACCIONES**

- Mover
- Copiar

Figura 7.1.33

Autor: Auditoría propia, herramienta usada: Trello.

- HU 34



The image shows a Trello card titled "HU - 34. Crear vista para ver las unidades de medida de una sede." (Create view to see measurement units of a site). The card is part of a "Sprint 4" list. It includes a description, a 100% progress bar, and a checklist of four items. On the right, there are sections for "AÑADIR A LA TARJETA" (Add to card) with options like "Miembros", "Etiquetas", "Checklist", "Vencimiento", "Adjunto", and "Portada"; and "POWER-UPS" with a "Conseguir Power-Ups" button and a link to "Actualizar el equipo".

**HU - 34. Crear vista para ver las unidades de medida de una sede.**  
en la lista [Sprint 4](#)

**Descripción** [Editar](#)

Yo como administrador del sistema requiero que la aplicación permita observar las unidades de medida que tiene la sede, para tener saber que unidades de medida se están trabajando en la sede..

**Criterios de aceptación** [Ocultar elementos completados](#) [Eliminar](#)

100%

- Se deben de observar todas las unidades de medida de la sede.
- Cada unidad de medida debe de mostrar la siguiente información:  
Descripción:
- Botón para crear una nueva unidad de medida.
- Botón para regresar al módulo de compras e inventario.

[Añada un elemento](#)

**AÑADIR A LA TARJETA**

- [Miembros](#)
- [Etiquetas](#)
- [Checklist](#)
- [Vencimiento](#)
- [Adjunto](#)
- [Portada](#)

**POWER-UPS**

[Conseguir Power-Ups](#)

Consiga Power-Ups ilimitados y mucho más.

[Actualizar el equipo](#)

Figura 7.1.34

Autor: Auditoría propia, herramienta usada: Trello.

- HU 35

**HU - 35. Implementar opciones al seleccionar una unidad de medida.**  
en la lista [Sprint 4](#)

**Descripción** Editar

Yo como administrador del sistema requiero que la aplicación permita seleccionar una unidad de medida para editarla o eliminarla.

Nota: El botón de eliminar estará activó siempre y cuando no se halla usado la unidad de medida en el inventario. Si se usó en el inventario de materias primas el botón de eliminar desaparecerá.

**Criterios de aceptación** Ocultar elementos completados Eliminar

100%

- ✓ Al dar clic en una unidad de medida, se debe de mostrar dos botones: Editar y Eliminar.
- ✓ Botón "Editar" para actualizar la información de la unidad de medida.
- ✓ Al dar clic en el botón de Editar, este debe de direccionar a la vista de Editar Unidad de medida.
- ✓ Al dar clic en el botón Eliminar, este debe de mostrar una ventana emergente de confirmación. Si se da aceptar el sistema debe de eliminar la unida de medida de la base de datos.
- ✓ Si se usó la unidad de medida en el inventario de materias primas, el botón de eliminar no se debe mostrar.
- ✓ Botón "Eliminar" para borrar la unidad de medida seleccionada.

Añada un elemento

**AÑADIR A LA TARJETA**

- Miembros
- Etiquetas
- Checklist
- Vencimiento
- Adjunto
- Portada

**POWER-UPS**

Conseguir Power-Ups

Consiga Power-Ups ilimitados y mucho más.

- Actualizar el equipo

**ACCIONES**

- Mover
- Copiar
- Convertir en plant...
- Seguir

Figura 7.1.35

Autor: Auditoría propia, herramienta usada: Trello.

- HU 36

**HU - 36. Editar o actualizar una unidad de medida.**  
en la lista [Sprint 4](#)

**Descripción** Edit

Yo como administrador del sistema requiero que la aplicación permita actualizar la información de una unidad de medida ya registrada en el sistema, para corregir información mal ingresada.

**Criterios de aceptación** Ocultar elementos completados Eliminar

100% 100%

- Se deben de mostrar la información de las unidades de medida con todos sus campos.
- Si se actualiza el campo de descripción, este solo aceptara una descripción mayor a cuatro caracteres.
- No se aceptarán campos en blanco(Nulos)
- Al dar clic en el botón de cancelar este debe de direccionar a la vista de ver unidades de medida
- Botón para actualizar la información.
- La unidad de medida debe de quedar con los datos actualizados correctamente.

Añade un elemento

**AÑADIR A LA TARJETA**

- Miembros
- Etiquetas
- Checklist
- Vencimiento
- Adjunto
- Portada

**POWER-UPS**

Conseguir Power-Ups

Consiga Power-Ups ilimitados y mucho más.

- Actualizar el equipo

**ACCIONES**

- Mover
- Copiar

Figura 7.1.36

Autor: Auditoría propia, herramienta usada: Trello.

- HU 37

**HU - 37. Crear vista del inventario bajo.**  
en la lista [Sprint 4](#)

**Descripción** Editar

Yo como administrador requiero que la aplicación permita observar los insumos o materias primas que se están agotando en el inventario, para poder tomar decisiones oportunas en las compras de nuevos insumos.

El inventario bajo, es todo insumo o materia prima que tiene una cantidad menor en la definida a la cantidad mínima de inventario. Cada materia prima tiene su propia cantidad mínima de inventario.

**Criterios de aceptación** Ocultar elementos completados Eliminar

100%

- Se deben de observar todas la materias primas que tienen una cantidad inferior al campo mínimo de inventario.
- Los campos que se debe de mostrar por cada materia prima son: Nombre, Unidad, Cantidad, Mínimo de inventario, Valor unidad, maneja inventario.
- Filtro para buscar por nombre de la materia prima.
- Botón para regresar al módulo de compras e inventarios.

Añada un elemento

**AÑADIR A LA TARJETA**

- Miembros
- Etiquetas
- Checklist
- Vencimiento
- Adjunto
- Portada

**POWER-UPS**

Conseguir Power-Ups

Consiga Power-Ups ilimitados y mucho más.

Actualizar el equipo

**ACCIONES**

- Mover
- Copiar

Figura 7.1.37

Autor: Auditoría propia, herramienta usada: Trello.

- HU 38

The image shows a Trello card titled "HU - 38. Implementar opciones al seleccionar una materia prima del inventario bajo." with a sub-header "en la lista [Sprint 4](#)". The card is organized into sections: "Descripción" with an "Editar" button, "Criterios de aceptación" with "Ocultar elementos completados" and "Eliminar" buttons, and a checklist with three items, all of which are marked as complete. A progress bar at the top of the checklist section shows 100% completion. On the right side, there are two utility sections: "AÑADIR A LA TARJETA" with buttons for "Miembros", "Etiquetas", "Checklist", "Vencimiento", "Adjunto", and "Portada"; and "POWER-UPS" with a "Conseguir Power-Ups" button and a promotional message "Consiga Power-Ups ilimitados y mucho más."

**HU - 38. Implementar opciones al seleccionar una materia prima del inventario bajo.**  
en la lista [Sprint 4](#)

**Descripción** Editar

Yo como administrador del sistema requiero que la aplicación permita seleccionar la materia prima reportada en el inventario bajo para ir a observar la información del proveedor de dicho insumo.

**Criterios de aceptación** Ocultar elementos completados Eliminar

100%

- Al dar clic en una materia prima, debe de mostrar un botón para ir a observar el proveedor de dicho insumo.
- Botón ver información proveedor.
- Al dar clic en el botón de "Ver información proveedor" debe de direccionar a la vista de observar proveedor.

Añada un elemento

**AÑADIR A LA TARJETA**

- Miembros
- Etiquetas
- Checklist
- Vencimiento
- Adjunto
- Portada

**POWER-UPS**

Conseguir Power-Ups

Consiga Power-Ups ilimitados y mucho más.

Figura 7.1.38

Autor: Auditoría propia, herramienta usada: Trello.

- HU 39

**HU - 39. Crear vista para ver los proveedores de la sede.**  
en la lista [Sprint 4](#)

**Descripción** Editar

Yo como administrador del sistema requiero que la aplicación permita observar las compras que se han realizado en la sede para poder conocer que se ha comprado..

**Criterios de aceptación** Ocultar elementos completados Eliminar

100%

- Se debe de mostrar todos lo proveedores que se hallan registrado en esa sede.
- Se debe de observar por proveedor la siguiente información: Descripción, Dirección y Correo.
- La información del proveedor observado tiene que coincidir con los datos que se ingresaron.
- Filtro para buscar un proveedor por el nombre del proveedor.
- Botón que permita crear una nuevo proveedor.
- Botón para volver al módulo de compras e inventario.

Añada un elemento

**AÑADIR A LA TARJETA**

- Miembros
- Etiquetas
- Checklist
- Vencimiento
- Adjunto
- Portada

**POWER-UPS**

Conseguir Power-Ups

Consiga Power-Ups ilimitados y mucho más.

Actualizar el equipo

**ACCIONES**

- Mover

Figura 7.1.39

Autor: Auditoría propia, herramienta usada: Trello.

- HU 40

**HU - 40. Ingresar un nuevo proveedor.**  
en la lista [Sprint 4](#)

**Descripción** Editar

Yo como administrador del sistema requiero que la aplicación permita ingresar un nuevo proveedor para tener la información de contacto de ellos.

**Criterios de aceptación** Ocultar elementos completados Eliminar

100%

- ✓ La aplicación permitirá ingresar la siguiente información: Descripción, Dirección y Correo.
- ✓ Todos los campos son obligatorios.
- ✓ El campo "Descripción", solo aceptará una descripción con al menos 4 caracteres.
- ✓ El campo "Dirección", solo aceptará una dirección con al menos 6 caracteres.
- ✓ El campo "Correo", solo aceptará un correo con al menos 6 caracteres. Además solo aceptará un correo con formato válido. El formato válido es que contenga texto + "@" + texto + .com
- ✓ Botón para crear el proveedor.
- ✓ Botón para cancelar el ingreso.
- ✓ Al dar clic en el botón "cancelar" este debe direccionar a la vista de ver proveedores.

Añada un elemento

**AÑADIR A LA TARJETA**

- Miembros
- Etiquetas
- Checklist
- Vencimiento
- Adjunto
- Portada

**POWER-UPS**

Conseguir Power-Ups

Consiga Power-Ups ilimitados y mucho más.

Actualizar el equipo

**ACCIONES**

- Mover
- Copiar
- Convertir en plant...
- Seguir

Figura 7.1.40

Autor: Auditoría propia, herramienta usada: Trello.

## Sprint 5

- HU 41

## HU - 41. Implementar opciones al seleccionar un proveedor.

en la lista [Sprint 5](#)

**Descripción** Editar

Yo como administrador del sistema requiero que la aplicación permita seleccionar un proveedor para poder editarlo, eliminarlo o ingresar a ver sus teléfonos.

Nota: El botón de eliminar estará activó siempre y cuando el proveedor seleccionado no tenga un teléfono ingresado. Si tiene un teléfono ingresado el botón eliminar desaparecerá.

**Criterios de aceptación** Ocultar elementos completados Eliminar

100%

- ✓ Al dar clic en un proveedor, se debe de mostrar tres botones: Editar, Eliminar y Teléfonos.
- ✓ Botón "Editar" para actualizar la información del proveedor.
- ✓ Al dar clic en el botón de Editar, este debe de direccionar a la vista de Editar proveedor.
- ✓ Botón "Eliminar" para borrar el proveedor seleccionado.
- ✓ Al dar clic en el botón Eliminar, este debe de mostrar una ventana emergente de confirmación. Si se da aceptar el sistema debe de eliminar el proveedor de la base de datos.
- ✓ Si el proveedor tiene un teléfono ingresado, el botón de eliminar no se debe mostrar.
- ✓ Botón "Teléfonos", para ingresar a ver los teléfonos del proveedor seleccionado.
- ✓ Al dar clic en el botón Teléfonos este debe de direccionar a la vista de ver teléfonos proveedor.

Añada un elemento

**AÑADIR A LA TARJETA**

- Miembros
- Etiquetas
- Checklist
- Vencimiento
- Adjunto
- Portada

**POWER-UPS**

Conseguir Power-Ups

Consiga Power-Ups ilimitados ¡ mucho más.

Actualizar el equipo

**ACCIONES**

- Mover
- Copiar
- Convertir en plant...
- Seguir
- Archivar
- Compartir

Figura 7.1.41

Autor: Auditoría propia, herramienta usada: Trello.

- HU 42

**HU - 42. Editar o actualizar información de un proveedor.**  
en la lista [Sprint 5](#)

**Descripción** Editar

Yo como administrador del sistema requiero que la aplicación permita actualizar la información de un proveedor ya ingresado en el sistema, para corregir información mal ingresada.

**Criterios de aceptación** Ocultar elementos completados Eliminar

100%

- ✓ Se debe de mostrar la información de los proveedores con todos sus campos.
- ✓ Si se actualiza el campo "Descripción", solo aceptará una descripción con al menos 4 caracteres.
- ✓ Si se actualiza el campo "Dirección", solo aceptará una dirección con al menos 6 caracteres.
- ✓ Si se actualiza el campo "Correo", solo aceptará un correo con al menos 6 caracteres. Además solo aceptará un correo con formato válido. El formato válido es que contenga texto + "@" + texto + .com
- ✓ Botón para actualizar la información.
- ✓ Botón para cancelar la transacción.
- ✓ Al dar clic en el botón "cancelar" este debe direccionar a la vista de ver proveedores.
- ✓ El proveedor seleccionado debe de quedar con los datos actualizados correctamente.

Añada un elemento

**AÑADIR A LA TARJETA**

- Miembros
- Etiquetas
- Checklist
- Vencimiento
- Adjunto
- Portada

**POWER-UPS**

Conseguir Power-Ups

Consiga Power-Ups ilimitados mucho más.

Actualizar el equipo

**ACCIONES**

- Mover
- Copiar
- Convertir en plant...
- Seguir
- Archivar

Figura 7.1.42

Autor: Auditoría propia, herramienta usada: Trello.

- HU 43

**HU - 43. Vista para ver los teléfonos de un proveedor.**  
en la lista [Sprint 5](#)

**Descripción** Editar

Yo como administrador del sistema requiero que la aplicación permita observar los teléfonos que tiene un proveedor, para poder consultar la información cada vez que se necesite.

**Criterios de aceptación** Ocultar elementos completados Eliminar

100%

- Se deben de observar todos los teléfonos que tenga el proveedor.
- Se debe de mostrar el siguiente campo: Teléfonos.
- Botón para regresar a la vista de proveedores.
- Botón para agregar un nuevo teléfono al proveedor.

Añada un elemento

**AÑADIR A LA TARJETA**

- Miembros
- Etiquetas
- Checklist
- Vencimiento
- Adjunto
- Portada

**POWER-UPS**

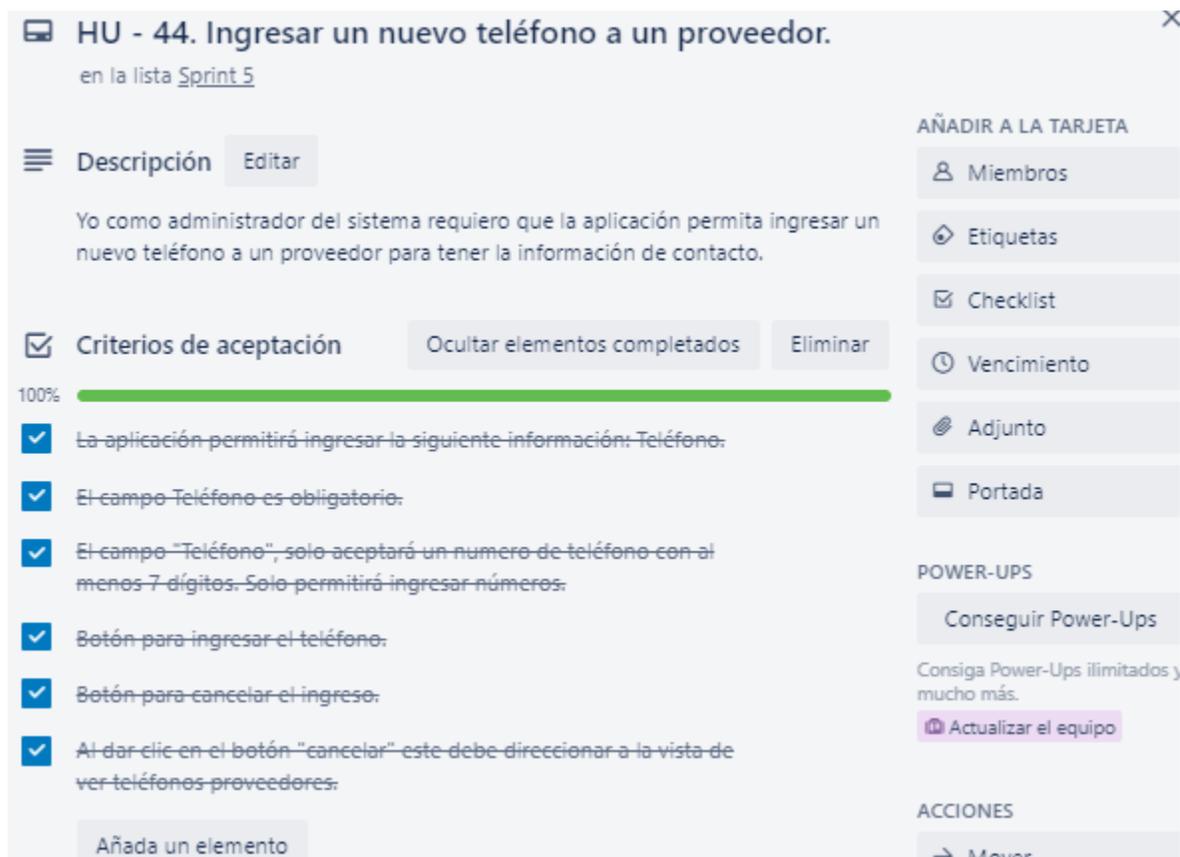
Conseguir Power-Ups

Consiga Power-Ups ilimitados y mucho más.

Figura 7.1.43

Autor: Auditoría propia, herramienta usada: Trello.

- HU 44



**HU - 44. Ingresar un nuevo teléfono a un proveedor.**  
en la lista [Sprint 5](#)

**Descripción** Editar

Yo como administrador del sistema requiero que la aplicación permita ingresar un nuevo teléfono a un proveedor para tener la información de contacto.

**Criterios de aceptación** Ocultar elementos completados Eliminar

100%

- ✓ La aplicación permitirá ingresar la siguiente información: Teléfono.
- ✓ El campo Teléfono es obligatorio.
- ✓ El campo "Teléfono", solo aceptará un número de teléfono con al menos 7 dígitos. Solo permitirá ingresar números.
- ✓ Botón para ingresar el teléfono.
- ✓ Botón para cancelar el ingreso.
- ✓ Al dar clic en el botón "cancelar" este debe direccionar a la vista de ver teléfonos proveedores.

Añada un elemento

**AÑADIR A LA TARJETA**

- Miembros
- Etiquetas
- Checklist
- Vencimiento
- Adjunto
- Portada

**POWER-UPS**

Conseguir Power-Ups

Consiga Power-Ups ilimitados y mucho más.

Actualizar el equipo

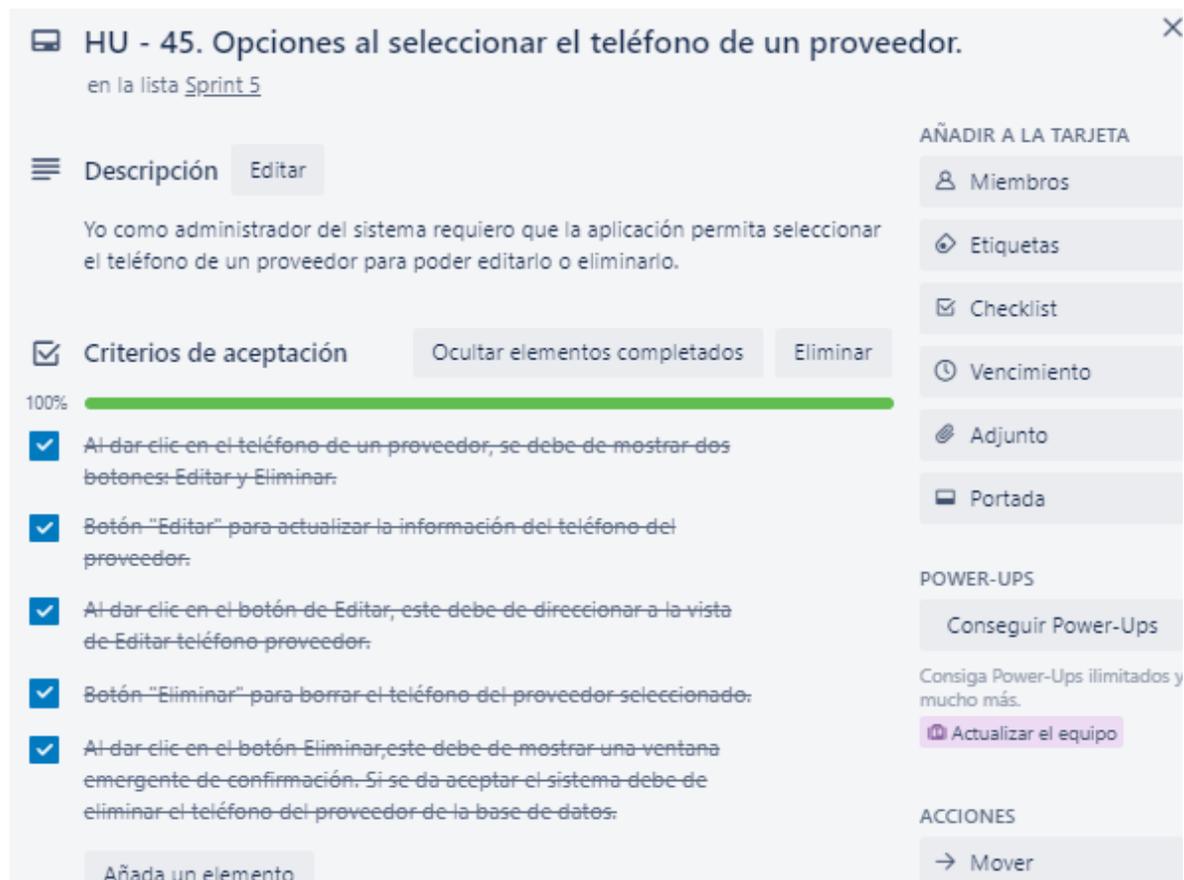
**ACCIONES**

- Mover

Figura 7.1.44

Autor: Auditoría propia, herramienta usada: Trello.

- HU 45



**HU - 45. Opciones al seleccionar el teléfono de un proveedor.**  
en la lista [Sprint 5](#)

**Descripción** [Editar](#)

Yo como administrador del sistema requiero que la aplicación permita seleccionar el teléfono de un proveedor para poder editarlo o eliminarlo.

**Criterios de aceptación** [Ocultar elementos completados](#) [Eliminar](#)

100%

- ✓ Al dar clic en el teléfono de un proveedor, se debe de mostrar dos botones: Editar y Eliminar.
- ✓ Botón "Editar" para actualizar la información del teléfono del proveedor.
- ✓ Al dar clic en el botón de Editar, este debe de direccionar a la vista de Editar teléfono proveedor.
- ✓ Botón "Eliminar" para borrar el teléfono del proveedor seleccionado.
- ✓ Al dar clic en el botón Eliminar, este debe de mostrar una ventana emergente de confirmación. Si se da aceptar el sistema debe de eliminar el teléfono del proveedor de la base de datos.

[Añada un elemento](#)

**AÑADIR A LA TARJETA**

- Miembros
- Etiquetas
- Checklist
- Vencimiento
- Adjunto
- Portada

**POWER-UPS**

[Conseguir Power-Ups](#)

Consiga Power-Ups ilimitados y mucho más.

[Actualizar el equipo](#)

**ACCIONES**

[→ Mover](#)

Figura 7.1.45

Autor: Auditoría propia, herramienta usada: Trello.

- HU 46

**HU - 46. Editar o actualizar el teléfono de un proveedor.**  
en la lista [Sprint 5](#)

**Descripción** [Editar](#)

Yo como administrador del sistema requiero que la aplicación permita actualizar la información de un teléfono de un proveedor ya ingresado en el sistema, para corregir información mal ingresada.

**Criterios de aceptación** [Ocultar elementos completados](#) [Eliminar](#)

100%

- ✓ Se deben de mostrar la información de los teléfonos con todos sus campos.
- ✓ Si se actualiza el campo "Teléfono", solo aceptará un número de teléfono con al menos 7 dígitos. Solo permitirá ingresar números.
- ✓ Botón para actualizar la información del teléfono.
- ✓ Botón para cancelar el proceso.
- ✓ Al dar clic en el botón "cancelar" este debe direccionar a la vista de ver teléfonos proveedores.
- ✓ El campo de "Teléfono" no puede estar vacío(Nulo).

[Añade un elemento](#)

**AÑADIR A LA TARJETA**

- Miembros
- Etiquetas
- Checklist
- Vencimiento
- Adjunto
- Portada

**POWER-UPS**

[Conseguir Power-Ups](#)

Consiga Power-Ups ilimitados y mucho más.

[Actualizar el equipo](#)

**ACCIONES**

- Mover
- Copiar

Figura 7.1.46

Autor: Auditoría propia, herramienta usada: Trello.

- HU 47

**HU - 47. Ingresar una nueva persona.**  
en la lista [Sprint 5](#)

**Descripción** Editar

Yo como administrador del sistema requiero que la aplicación permita ingresar una nueva persona, para guardar sus datos en el sistema.

**Criterios de aceptación** Ocultar elementos completados Eliminar

100%

- ✓ La aplicación debe permitir ingresar la siguiente información: Nombre, Primer Apellido, Segundo Apellido, Tipo de teléfono, Número de identificación, Dirección, Sexo, Tipo de persona, Fecha ingreso;
- ✓ El campo "Nombre" solo aceptará un nombre con al menos 4 caracteres;
- ✓ El campo "Primer apellido" solo aceptará un apellido con al menos 4 caracteres;
- ✓ El campo "Segundo apellido" solo aceptará un apellido con al menos 4 caracteres;
- ✓ El sistema debe de mostrar los tipos de documentos disponibles y permitir elegir una de ellos;
- ✓ El campo "Número de identificación" solo aceptará un número de id con al menos 5 caracteres;
- ✓ El campo "Dirección" solo aceptará una dirección con al menos 5 caracteres;
- ✓ El sistema debe de mostrar el sexo a elegir, M para masculino y F para femenino y permitir elegir uno de ellos;
- ✓ El sistema debe de mostrar los Tipos de personas disponibles y permitir elegir una de ellos;
- ✓ El campo Fecha de ingreso, sólo aceptará la fecha en formato dd/mm/aaaa
- ✓ Botón para crear o guardar la persona. Debe de guardar la persona en estado true por defecto;
- ✓ Botón para cancelar el ingreso de la persona;
- ✓ Al dar clic en el botón cancelar, este debe de direccionar a la vista de ver personas;

Añada un elemento

**AÑADIR A LA TARJETA**

- Miembros
- Etiquetas
- Checklist
- Vencimiento
- Adjunto
- Portada

**POWER-UPS**

Consiga Power-Ups ilimitados mucho más.

Actualizar el equipo

**ACCIONES**

- Mover
- Copiar
- Convertir en plant...
- Seguir
- Archivar
- Compartir

Figura 7.1.47

Autor: Auditoría propia, herramienta usada: Trello.

- HU 48

**HU - 48. Vista para ver personas.**  
en la lista [Sprint 5](#)

**Descripción** [Editar](#)

Yo como administrador del sistema requiero que la aplicación permita observar las personas que se han ingresado a la sede, para consultar la información de las personas.

**Criterios de aceptación** [Ocultar elementos completados](#) [Eliminar](#)

100%

- Se deben de observar las personas de esa sede.
- Se debe de observar por persona la siguiente información: Nombre, Primer apellido, Segundo apellido, Tipo de documento, Número de identificación, estado.
- La información de la persona observada tiene que coincidir con la persona que se ingreso.
- Filtro para buscar una persona por su nombre.
- Botón que permita crear una nueva persona.
- Botón para volver al módulo de talento humano.

[Añada un elemento](#)

**AÑADIR A LA TARJETA**

- [Miembros](#)
- [Etiquetas](#)
- [Checklist](#)
- [Vencimiento](#)
- [Adjunto](#)
- [Portada](#)

**POWER-UPS**

[Conseguir Power-Ups](#)

Consiga Power-Ups ilimitados y mucho más.

[Actualizar el equipo](#)

**ACCIONES**

- [Mover](#)
- [Copiar](#)

Figura 7.1.48

Autor: Auditoría propia, herramienta usada: Trello.

- HU 49

**HU - 49. Implementar opciones al seleccionar una persona.**  
en la lista [Sprint 5](#)

**Descripción** Editar

Yo como administrador del sistema requiero que la aplicación permita seleccionar una persona para poder realizar las siguientes acciones:

- Ver información detallada de la persona seleccionada.
- Editar o actualizar la información de la persona seleccionada.
- Eliminar la personas seleccionada.
- Ver los teléfonos de esa persona seleccionada.

Nota: El botón de eliminar estará activo siempre y cuando no tenga un teléfono registrado. Si se registró un teléfono el botón de eliminar desaparecerá.

**Criterios de aceptación** Ocultar elementos completados Eliminar

100%

- ✓ Al dar clic en una persona se debe de mostrar dos botones: Ver información, Editar, Eliminar y Teléfonos.
- ✓ Botón "Ver información" para observar la información detallada de una persona.
- ✓ Al dar clic en el botón "Ver información" debe de direccionar a la vista de ver información de la persona.
- ✓ Botón "Editar" para ir a actualizar la información de una persona.
- ✓ Al dar clic en el botón de Editar, este debe de direccionar a la vista de Editar persona.
- ✓ Botón "Eliminar" para borrar la persona seleccionada.
- ✓ Al dar clic en el botón Eliminar, este debe de mostrar una ventana emergente de confirmación. Si se da aceptar el sistema debe de eliminar la persona del sistema.
- ✓ Si la persona tiene un teléfono, el botón de eliminar no se debe mostrar.

Añada un elemento

**AÑADIR A LA TARJETA**

- Miembros
- Etiquetas
- Checklist
- Vencimiento
- Adjunto
- Portada

**POWER-UPS**

Conseguir Power-Ups

Consiga Power-Ups ilimitados y mucho más.

- Actualizar el equipo

**ACCIONES**

- Mover
- Copiar
- Convertir en plant...
- Seguir
- Archivar
- Compartir

Figura 7.1.49

Autor: Auditoría propia, herramienta usada: Trello.

- HU 50

**HU - 50. Editar o actualizar información de una persona.**  
en la lista [Sprint 5](#)

**Descripción** Editar

Yo como administrador del sistema requiero que la aplicación permita actualizar la información de una persona, para corregir datos mal ingresados.

**Criterios de aceptación** Ocultar elementos completados Eliminar

100%

- Se debe de mostrar la siguiente información: Nombre, Primer Apellido, Segundo Apellido, Tipo de teléfono, Número de identificación, Dirección, Sexo, Tipo de persona, Fecha ingreso, estado.
- Si se actualiza el campo "Nombre" solo aceptará un nombre con al menos 4 caracteres.
- Si se actualiza el campo "Primer apellido" solo aceptará un apellido con al menos 4 caracteres.
- Si se actualiza el campo "Segundo apellido" solo aceptará un apellido con al menos 4 caracteres.
- El sistema debe de mostrar los tipos de documentos disponibles y permitir elegir una de ellos para cambiarlo.
- Si se actualiza el campo "Número de identificación" solo aceptará un número de id con al menos 5 caracteres.
- Si se actualiza el campo "Dirección" solo aceptará una dirección con al menos 5 caracteres.
- El sistema debe de mostrar el sexo a elegir, M para masculino y F para femenino y permitir elegir uno de ellos para cambiarlo.
- Si se actualiza el campo Fecha de ingreso, sólo aceptará la fecha en formato dd/mm/aaaa
- Botón para actualizar la información de la persona.
- Botón para cancelar el proceso.
- Al dar clic en el botón cancelar, este debe de direccionar a la vista de ver personas.
- La información editada debe de quedar actualizada correctamente.

Añada un elemento

**AÑADIR A LA TARJETA**

- Miembros
- Etiquetas
- Checklist
- Vencimiento
- Adjunto
- Portada

**POWER-UPS**

Conseguir Power-Ups

Consiga Power-Ups ilimitados y mucho más.

Actualizar el equipo

**ACCIONES**

- Mover
- Copiar
- Convertir en plant...
- Seguir
- Archivar
- Compartir

Figura 7.1.50

Autor: Auditoría propia, herramienta usada: Trello.

- HU 51

**HU - 51. Ingresar teléfono a una persona.**  
en la lista [Sprint 5](#)

**Descripción** [Editar](#)

Yo como administrador del sistema requiero que la aplicación permita ingresar un teléfono a una persona ya registrada, para poder guardar en el sistema su teléfono.

**Criterios de aceptación** [Ocultar elementos completados](#) [Eliminar](#)

100%

- El sistema debe de permitir ingresar la siguiente información: Tipo de teléfono, Teléfono.
- En el campo de tipo de teléfono se debe de mostrar y permitir elegir uno de los tipos de teléfonos que estén registrados en la sede.
- El campo "Teléfono", solo aceptará un teléfono con al menos 7 caracteres.
- Botón para Guardar el nuevo teléfono.
- Botón para cancelar el ingreso.
- Al dar clic en cancelar, este debe de direccionar a la vista del ver teléfonos de persona.

[Añada un elemento](#)

**AÑADIR A LA TARJETA**

- [Miembros](#)
- [Etiquetas](#)
- [Checklist](#)
- [Vencimiento](#)
- [Adjunto](#)
- [Portada](#)

**POWER-UPS**

[Conseguir Power-Ups](#)

Consiga Power-Ups ilimitados mucho más.

[Actualizar el equipo](#)

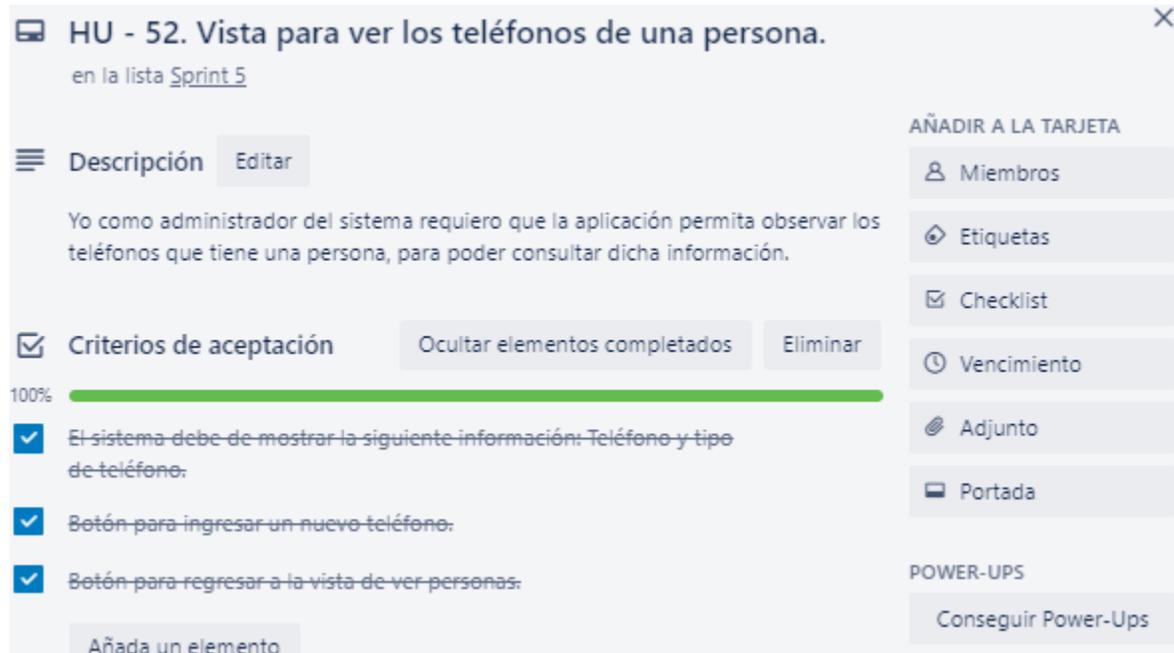
**ACCIONES**

- [→ Mover](#)
- [Copiar](#)

Figura 7.1.51

Autor: Auditoría propia, herramienta usada: Trello.

- HU 52



The image shows a Trello card titled "HU - 52. Vista para ver los teléfonos de una persona." located in the "Sprint 5" list. The card is divided into several sections:

- Descripción:** A text block stating, "Yo como administrador del sistema requiero que la aplicación permita observar los teléfonos que tiene una persona, para poder consultar dicha información." There is an "Editar" button next to it.
- Criterios de aceptación:** A section with a 100% progress bar and a green bar. It contains three items, each with a blue checkmark:
  - El sistema debe de mostrar la siguiente información: Teléfono y tipo de teléfono.
  - Botón para ingresar un nuevo teléfono.
  - Botón para regresar a la vista de ver personas.Buttons for "Ocultar elementos completados" and "Eliminar" are present.
- AÑADIR A LA TARJETA:** A sidebar on the right with buttons for "Miembros", "Etiquetas", "Checklist", "Vencimiento", "Adjunto", and "Portada".
- POWER-UPS:** A button labeled "Conseguir Power-Ups" at the bottom right.

Figura 7.1.52

Autor: Auditoría propia, herramienta usada: Trello.

## Sprint 6

- HU 53

**HU- 53. Implementar opciones al seleccionar un teléfono de una personas.**  
en la lista [Sprint 6](#)

**Descripción** Editar

yo como administrador del sistema requiero que el sistema permita seleccionar un teléfono de una persona para:

- Actualizar la información del teléfono seleccionado.
- Eliminar el teléfono seleccionado.

**Criterios de aceptación** Ocultar elementos completados Eliminar

100%

- ✓ Al dar clic en un teléfono se debe de mostrar las siguiente opciones: Editar y Eliminar.
- ✓ Al dar clic en el botón "Editar", este debe de direccionar a ala vista de Editar teléfono persona.
- ✓ Botón "Eliminar" para borrar el teléfono de la persona seleccionada.
- ✓ Al dar clic en el botón Eliminar, este debe de mostrar una ventana emergente de confirmación. Si se da aceptar el sistema debe de eliminar el teléfono de la persona del sistema.

Añada un elemento

**AÑADIR A LA TARJETA**

- Miembros
- Etiquetas
- Checklist
- Vencimiento
- Adjunto
- Portada

**POWER-UPS**

Conseguir Power-Ups

Consiga Power-Ups ilimitados y mucho más.

Actualizar el equipo

**ACCIONES**

→ Mover

Figura 7.1.53

Autor: Auditoría propia, herramienta usada: Trello.

- HU 54

**HU - 54. Editar o actualizar información de un teléfono de una persona.** ✕

en la lista [Sprint 6](#)

**Descripción** Editar

Yo como administrador del sistema requiero que la aplicación permita actualizar la información de un teléfono de la persona seleccionada, para poder corregir información o actualizarla.

**Criterios de aceptación** Ocultar elementos completados Eliminar

100%

- ✓ Se debe de mostrar la siguiente información: Tipo de teléfono y teléfono:
- ✓ En el campo de tipo de teléfono se debe de mostrar y permitir elegir uno de los tipos de teléfonos que estén registrados en la sede:
- ✓ Si se actualiza el campo "Teléfono", solo aceptará un teléfono con al menos 7 caracteres:
- ✓ Botón para Guardar los datos actualizados:
- ✓ Botón para cancelar el proceso:
- ✓ Al dar clic en cancelar, este debe de direccionar a la vista del ver teléfonos de persona:
- ✓ Los datos actualizados deben de verse reflejados al ver de nuevo la información del teléfono de la persona seleccionada:

Añada un elemento

**AÑADIR A LA TARJETA**

- Miembros
- Etiquetas
- Checklist
- Vencimiento
- Adjunto
- Portada

**POWER-UPS**

Conseguir Power-Ups

Consiga Power-Ups ilimitados y mucho más.

- Actualizar el equipo

**ACCIONES**

- Mover
- Copiar
- Convertir en plant...

Figura 7.1.54

Autor: Auditoría propia, herramienta usada: Trello.

- HU 55

**HU - 55. Ingresar un nuevo empleado.**  
en la lista [Sprint 6](#)

**Descripción** [Editar](#)

Yo como administrador del sistema requiero que la aplicación me permita ingresar un empleado , para poder darle un Rol/Cargo en la empresa.

Nota: Sólo se puede crear un empleado, a una persona que se encuentre creada en el sistema y se encuentre en estado activa.

**Criterios de aceptación** [Ocultar elementos completados](#) [Eliminar](#)

100%

- ✓ La aplicación debe permitir ingresar la siguiente información: Persona, Rol, Correo, Salario, Eps, ARL, Fechas de ingreso.
- ✓ En el campo "Persona", el sistema debe de permitir elegir una persona registrada en el sistema que cumpla con lo siguiente: Tienes este activa y que aún no se empleado.
- ✓ En el campo "Rol", el sistema debe de permitir elegir los roles que se encuentran registrados en el sistema.
- ✓ El campo "Correo", sólo permitirá ingresar un correo que cumpla con el siguiente formato: caracteres + @ + caracteres + .com
- ✓ El campo "Salario", solo aceptará un salario que tenga por lo menos tres dígitos, además solo se permiten números y decimales.
- ✓ El campo "EPS", solo aceptará una EPS con al menos 3 caracteres.
- ✓ El campo "ARL", solo aceptará una ARL con al menos 3 caracteres.
- ✓ El campo "Fecha de ingreso", sólo aceptará el siguiente formato: dd/mm/aaaa.
- ✓ Todos los campos son obligatorios.
- ✓ Botón para crear el empleado. Debe de crear el empleado en estado True por defecto.
- ✓ Botón para cancelar el ingreso. Al dar clic en este botón, debe de direccionar a a la vista de ver empleados.

[Añada un elemento](#)

**AÑADIR A LA TARJETA**

- Miembros
- Etiquetas
- Checklist
- Vencimiento
- Adjunto
- Portada

**POWER-UPS**

[Conseguir Power-Ups](#)

Consiga Power-Ups ilimitados y mucho más.

[Actualizar el equipo](#)

**ACCIONES**

- Mover
- Copiar
- Convertir en plant...
- Seguir
- Archivar
- Compartir

Figura 7.1.55

Autor: Auditoría propia, herramienta usada: Trello.

- HU 56

**HU - 56. Vista para ver los empleados.**  
en la lista [Sprint 6](#)

**Descripción** Editar

Yo como administrador del sistema requiero que la aplicación permita ver los empleados que tiene la sede, ya sean activo e inactivos. Para poder saber el cargo, su estado y su nombre.

**Criterios de aceptación** Ocultar elementos completados Eliminar

100%

- El sistema debe demostrar la siguiente información por cada empleado: Nombre, Rol/Cargo y Estado.
- El nombre del empleado debe de coincidir con el nombre de la persona a la cual se ingresó como empleado.
- Filtro para buscar un empleado por su nombre.
- Botón para ingresar un nuevo empleado.
- Botón para regresar al módulo e talento humano.

Añada un elemento

**AÑADIR A LA TARJETA**

- Miembros
- Etiquetas
- Checklist
- Vencimiento
- Adjunto
- Portada

**POWER-UPS**

Conseguir Power-Ups

Consiga Power-Ups ilimitados y mucho más.

Actualizar el equipo

**ACCIONES**

Figura 7.1.56

Autor: Auditoría propia, herramienta usada: Trello.

- HU 57

**HU - 57. Implementar Opciones al seleccionar un empleado.**  
en la lista [Sprint 6](#)

**Descripción** Editar

Yo como administrador del sistema requiero que la aplicación permita seleccionar un empleado para:

- Ver información detallada del empleado seleccionado.
- Editar o actualizar la información del empleado seleccionado.
- Eliminar el empleado seleccionado.

Nota: Solo se podrá eliminar un empleado que no tenga un usuario y contraseña asignado, de lo contrario el botón no se mostrará.

**Criterios de aceptación** Ocultar elementos completados Eliminar

100%

- ✓ Al seleccionar un empleado, se debe de mostrar las siguientes opciones: Ver empleado y Editar.
- ✓ Botón para actualizar la información de un empleado.
- ✓ Al dar clic en el botón de editar, este debe de direccionar a la vista de Editar empleado.
- ✓ Botón para eliminar el empleado.
- ✓ Al dar clic en el botón Eliminar, este debe de mostrar una ventana emergente de confirmación. Si se da aceptar el sistema debe de eliminar el empleado del sistema.
- ✓ Si el empleado tiene usuario y contraseña, el botón de eliminar no se debe mostrar.
- ✓ Botón para ver información detallada del empleado.
- ✓ Al dar clic en el botón de ver información, este debe de direccionar a la vista de ver información de empleado.

Añada un elemento

**AÑADIR A LA TARJETA**

- Miembros
- Etiquetas
- Checklist
- Vencimiento
- Adjunto
- Portada

**POWER-UPS**

Conseguir Power-Ups

Consiga Power-Ups ilimitados mucho más.

- Actualizar el equipo

**ACCIONES**

- Mover
- Copiar
- Convertir en plant...
- Seguir
- Archivar
- Compartir

Figura 7.1.57

Autor: Auditoría propia, herramienta usada: Trello.

- HU 58

## HU - 58. Editar o actualizar información de un empleado.

✕

en la lista [Sprint 6](#)

☰ Descripción
Editar

Yo cómo administrador del sistema requiero que la aplicación permita actualizar la información de un empleado, para corregir datos mal ingresados o para tener la información actual de los empleados.

**Criterios de aceptación**

Ocultar elementos completados
Eliminar

100%

- Se debe de mostrar la siguiente información para editar: Rol, Correo, Salario, EPS, ARL, Fecha de ingreso, Estado.
- En el campo "Rol", el sistema debe de permitir elegir los roles que se encuentran registrados en el sistema por si se desea cambiar.
- Si se actualiza el campo "Correo", sólo permitirá ingresar un correo que cumpla con el siguiente formato: caracteres + @ + caracteres + .com
- Si se actualiza el campo "Salario", solo aceptará un salario que tenga por lo menos tres dígitos, además solo se permiten números y decimales.
- Si se actualiza el campo "EPS", solo aceptará una EPS con al menos 3 caracteres.
- Si se actualiza el campo "ARL", solo aceptará una ARL con al menos 3 caracteres.
- Si se actualiza el campo "Fecha de ingreso", sólo aceptará el siguiente formato: dd/mm/aaaa.
- El sistema debe de permitir elegir el estado del empleado para cambiarlo si se desea, los estaos son lo siguientes: Activo, Inactivo.
- Ningún campo puede estar vacío(Nulo).
- Botón para actualizar la información del empleado.
- Botón para cancelar el proceso. Al dar clic en este botón, debe de direccionar a a la vista de ver empleados.

Añada un elemento

AÑADIR A LA TARJETA

- Miembros
- Etiquetas
- Checklist
- Vencimiento
- Adjunto
- Portada

POWER-UPS

Conseguir Power-Ups

Consiga Power-Ups ilimitados y mucho más.

Actualizar el equipo

ACCIONES

- Mover
- 📄 Copiar
- 📄 Convertir en plant...
- 👁️ Seguir
- 📁 Archivar
- 🔗 Compartir

Figura 7.1.58

Autor: Auditoría propia, herramienta usada: Trello.

- HU 59

**HU - 59. Ver información detallada de un empleado.**  
en la lista [Sprint 6](#)

**Descripción** Editar

Yo como administrador del sistema requiero que la aplicación permita seleccionar un empleado e ingresar a su información, para ver la información detallada de ese empleado

**Criterios de aceptación** Ocultar elementos completados Eliminar

100%

- Se debe de observar la siguiente información: Nombre completo, Rol/Cargo, Identificación, Correo, Salario, EPS, ARL, Estado, Fecha de ingreso.
- Botón para regresar a la vista de ver empleados.

Añada un elemento

**AÑADIR A LA TARJETA**

- Miembros
- Etiquetas
- Checklist
- Vencimiento
- Adjunto
- Portada

**POWER-UPS**

Conseguir Power-Ups

Consigue Power-Ups ilimitados y

Figura 7.1.59

Autor: Auditoría propia, herramienta usada: Trello.

- HU 60

**HU -60. Ingresar un nuevo tipo de persona.**  
en la lista [Sprint 6](#)

**Descripción** Editar

Yo como administrador del sistema requiero que la aplicación permita ingresar un nuevo tipo de persona, para poder asignárselo al ingresar un nueva persona.

**Criterios de aceptación** Ocultar elementos completados Eliminar

100%

- El sistema debe de permitir ingresar la siguiente información:  
Descripción:
- El campo "Descripción", solo aceptará una descripción con al menos 2 caracteres.
- El campo "Descripción" es obligatorio.
- Botón para Guardar el nuevo tipo de persona.
- Botón para cancelar el ingreso.
- Al dar clic en cancelar, este debe de direccionar a la vista del ver tipos de persona.

Añada un elemento

**AÑADIR A LA TARJETA**

- Miembros
- Etiquetas
- Checklist
- Vencimiento
- Adjunto
- Portada

**POWER-UPS**

Conseguir Power-Ups

Consiga Power-Ups ilimitados y mucho más.

Actualizar el equipo

**ACCIONES**

- Mover

Figura 7.1.60

Autor: Auditoría propia, herramienta usada: Trello.

- HU 61

**HU - 61. Vista para ver los tipos de persona de una sede.**  
en la lista [Sprint 6](#)

**Descripción** Editar

Yo como administrador del sistema requiero que la aplicación permita observar los tipos de persona que se han ingresado al sistema, para poder ver que tipos de personas usa la sede para asignar a una persona.

**Criterios de aceptación** Ocultar elementos completados Eliminar

100%

- El sistema debe de mostrar la siguiente información: Tipo de persona;
- Botón para ingresar un nuevo tipo de persona;
- Botón para regresar a la vista del módulo de talento humano;

Añada un elemento

**AÑADIR A LA TARJETA**

- Miembros
- Etiquetas
- Checklist
- Vencimiento
- Adjunto
- Portada

**POWER-UPS**

Conseguir Power-Ups

Consiga Power-Ups ilimitados y mucho más.

Figura 7.1.61

Autor: Auditoría propia, herramienta usada: Trello.

- HU 62

**HU - 62. Implementar opciones al seleccionar un tipo de persona.**  
en la lista [Sprint 6](#)

**Descripción** [Editar](#)

yo como administrador del sistema requiero que el sistema permita seleccionar un tipo de persona para:

- Actualizar la información del tipo de persona seleccionado.
- Eliminar el tipo de persona seleccionado.

**Criterios de aceptación** [Ocultar elementos completados](#) [Eliminar](#)

100%

- ✓ Al dar clic en un tipo de persona se debe de mostrar las siguiente opciones: Editar y Eliminar.
- ✓ Al dar clic en el botón "Editar", este debe de direccionar a ala vista de Editar tipo de persona.
- ✓ Botón "Eliminar" para borrar el tipo de persona seleccionado.
- ✓ Al dar clic en el botón Eliminar,este debe de mostrar una ventana emergente de confirmación. Si se da aceptar el sistema debe de eliminar el tipo de persona del sistema.

[Añada un elemento](#)

**AÑADIR A LA TARJETA**

- Miembros
- Etiquetas
- Checklist
- Vencimiento
- Adjunto
- Portada

**POWER-UPS**

Conseguir Power-Ups

Consiga Power-Ups ilimitados y mucho más.

[Actualizar el equipo](#)

**ACCIONES**

- Mover

Figura 7.1.62

Autor: Auditoría propia, herramienta usada: Trello.

- HU 63

**HU - 63. Editar o actualizar la información de un tipo de persona.**  
en la lista [Sprint 6](#)

**Descripción** Editar

Yo como administrador del sistema requiero que la aplicación permita actualizar la información de un tipo de persona seleccionada, para poder corregir información o actualizarla.

**Criterios de aceptación** Ocultar elementos completados Eliminar

100%

- ✓ Se debe de mostrar la siguiente información: Descripción:
- ✓ Si se actualiza el campo "Descripción", solo aceptará una descripción con al menos 2 caracteres.
- ✓ El campo "Descripción" no puede estar vacío(Nulo).
- ✓ Botón para cancelar el proceso.
- ✓ Al dar clic en cancelar, este debe de direccionar a la vista del ver tipos de persona.
- ✓ Botón para actualizar lo datos.
- ✓ Los datos actualizados deben de versen reflejados al ver de nuevo la información del tipo de persona seleccionada.

Añada un elemento

**AÑADIR A LA TARJETA**

- Miembros
- Etiquetas
- Checklist
- Vencimiento
- Adjunto
- Portada

**POWER-UPS**

Consiga Power-Ups ilimitados y mucho más.

- Actualizar el equipo

**ACCIONES**

- Mover
- Copiar
- Convertir en plant...

Figura 7.1.63

Autor: Auditoría propia, herramienta usada: Trello.

- HU 64

**HU -64. Ingresar un nuevo tipo de documento.**  
en la lista [Sprint 6](#)

**Descripción** Editar

Yo como administrador del sistema requiero que la aplicación permita ingresar un nuevo tipo de documento, para poder asignárselo al ingresar un nueva persona.

**Criterios de aceptación** Ocultar elementos completados Eliminar

100%

- El sistema debe de permitir ingresar la siguiente información:  
Descripción:
- El campo "Descripción", solo aceptará una descripción con al menos 2-caracteres.
- El campo "Descripción" es obligatorio.
- Botón para Guardar el nuevo tipo de documento.
- Botón para cancelar el ingreso.
- Al dar clic en cancelar, este debe de direccionar a la vista del ver tipos de documentos.

Añada un elemento

**AÑADIR A LA TARJETA**

- Miembros
- Etiquetas
- Checklist
- Vencimiento
- Adjunto
- Portada

**POWER-UPS**

Conseguir Power-Ups

Consiga Power-Ups ilimitados y mucho más.

- Actualizar el equipo

**ACCIONES**

- Mover

Figura 7.1.64

Autor: Auditoría propia, herramienta usada: Trello.

- HU 65

**HU - 65. Vista para ver los tipos de documentos de una sede.**  
en la lista [Sprint 6](#)

**Descripción** Editar

Yo como administrador del sistema requiero que la aplicación permita observar los tipos de documentos que se han ingresado al sistema, para poder ver que tipos de documentos usa la sede para asignar a una persona.

**Criterios de aceptación** Ocultar elementos completados Eliminar

100%

- El sistema debe de mostrar la siguiente información: Tipo de documento:
- Botón para ingresar un nuevo tipo de documento:
- Botón para regresar a la vista del módulo de talento humano:

Añada un elemento

**AÑADIR A LA TARJETA**

- Miembros
- Etiquetas
- Checklist
- Vencimiento
- Adjunto
- Portada

**POWER-UPS**

Conseguir Power-Ups

Consiga Power-Ups ilimitados y

Figura 7.1.65

Autor: Auditoría propia, herramienta usada: Trello.

- HU 66

**HU - 66. Implementar opciones al seleccionar un tipo de documento.**  
en la lista [Sprint 6](#)

**Descripción** Editar

yo como administrador del sistema requiero que el sistema permita seleccionar un tipo de documento para:

- Actualizar la información del tipo de documento seleccionado.
- Eliminar el tipo de documento seleccionado.

**✓ Criterios de aceptación** Ocultar elementos completados Eliminar

100%

- ✓ Al dar clic en un tipo de documento se debe de mostrar las siguientes opciones: Editar y Eliminar.
- ✓ Al dar clic en el botón "Editar", este debe de direccionar a la vista de Editar tipo de documento.
- ✓ Botón "Eliminar" para borrar el tipo de documento seleccionado.
- ✓ Al dar clic en el botón Eliminar, este debe de mostrar una ventana emergente de confirmación. Si se da aceptar el sistema debe de eliminar el tipo de documento del sistema.

Añada un elemento

**AÑADIR A LA TARJETA**

- Miembros
- Etiquetas
- Checklist
- Vencimiento
- Adjunto
- Portada

**POWER-UPS**

Consiga Power-Ups ilimitados y mucho más.

- Actualizar el equipo

**ACCIONES**

- Mover

Figura 7.1.66

Autor: Auditoría propia, herramienta usada: Trello.

- HU 67

**HU - 67. Editar o actualizar la información de un tipo de persona.**  
en la lista [Sprint 6](#)

**Descripción** Editar

Yo como administrador del sistema requiero que la aplicación permita actualizar la información de un tipo de documento seleccionado, para poder corregir información o actualizarla.

**Criterios de aceptación** Ocultar elementos completados Eliminar

100% 100%

- Se debe de mostrar la siguiente información: Descripción.
- Si se actualiza el campo "Descripción", solo aceptará una descripción con al menos 2 caracteres.
- El campo "Descripción" no puede estar vacío(Nulo).
- Botón para cancelar el proceso.
- Al dar clic en cancelar, este debe de direccionar a la vista del ver tipos de documentos.
- Botón para actualizar lo datos.
- Los datos actualizados deben de verse reflejados al ver de nuevo la información del tipo de documento seleccionado.

Añada un elemento

**AÑADIR A LA TARJETA**

- Miembros
- Etiquetas
- Checklist
- Vencimiento
- Adjunto
- Portada

**POWER-UPS**

Conseguir Power-Ups

Consiga Power-Ups ilimitados y mucho más.

- Actualizar el equipo

**ACCIONES**

- Mover
- Copiar

Figura 7.1.67

Autor: Auditoría propia, herramienta usada: Trello.

## Sprint 7

- HU 68

The image shows a Trello card titled "HU -68. Ingresar un nuevo tipo de teléfono." located in the "Sprint 7" list. The card contains a description, acceptance criteria, and various utility buttons.

**Descripción** Editar

Yo como administrador del sistema requiero que la aplicación permita ingresar un nuevo tipo de teléfono, para poder asignárselo al ingresar un teléfono a una persona.

**Criterios de aceptación** Ocultar elementos completados Eliminar

100% Progress bar

- El sistema debe de permitir ingresar la siguiente información: Descripción.
- El campo "Descripción", solo aceptará una descripción con al menos 2 caracteres.
- El campo "Descripción" es obligatorio.
- Botón para Guardar el nuevo tipo de teléfono.
- Botón para cancelar el ingreso.
- Al dar clic en cancelar, este debe de direccionar a la vista del ver tipos de teléfonos.

Añada un elemento

**AÑADIR A LA TARJETA**

- Miembros
- Etiquetas
- Checklist
- Vencimiento
- Adjunto
- Portada

**POWER-UPS**

Conseguir Power-Ups

Consiga Power-Ups ilimitados y mucho más.

Actualizar el equipo

**ACCIONES**

- Mover
- Copiar

Figura 7.1.68

Autor: Auditoría propia, herramienta usada: Trello.

- HU 69

The image shows a Trello card titled "HU - 69. Vista para ver los tipos de teléfonos de una sede." with a sub-header "en la lista [Sprint 7](#)". The card is organized into sections: "Descripción" with an "Editar" button and a text block; "Criterios de aceptación" with a 100% progress bar, "Ocultar elementos completados", and "Eliminar" buttons, followed by three checked items; and a "POWER-UPS" section with a "Conseguir Power-Ups" button. A right-hand sidebar offers options to "AÑADIR A LA TARJETA" such as "Miembros", "Etiquetas", "Checklist", "Vencimiento", "Adjunto", and "Portada".

**HU - 69. Vista para ver los tipos de teléfonos de una sede.**  
en la lista [Sprint 7](#)

**Descripción** Editar

Yo como administrador del sistema requiero que la aplicación permita observar los tipos de teléfonos que se han ingresado al sistema, para poder ver que tipos de teléfonos usa la sede para asignar a un teléfono de una persona.

**Criterios de aceptación** Ocultar elementos completados Eliminar

100%

- El sistema debe de mostrar la siguiente información: Tipo de teléfono.
- Botón para ingresar un nuevo tipo de teléfono.
- Botón para regresar a la vista del módulo de talento humano.

Añada un elemento

**AÑADIR A LA TARJETA**

- Miembros
- Etiquetas
- Checklist
- Vencimiento
- Adjunto
- Portada

**POWER-UPS**

Conseguir Power-Ups

Consiga Power-Ups ilimitados y mucho más.

Figura 7.1.69

Autor: Auditoría propia, herramienta usada: Trello.

- HU 70

**HU - 70. Implementar opciones al seleccionar un tipo de teléfono.**  
en la lista [Sprint 7](#)

**Descripción** Editar

yo como administrador del sistema requiero que el sistema permita seleccionar un tipo de teléfono para:

- Actualizar la información del tipo de teléfono seleccionado.
- Eliminar el tipo de teléfono seleccionado.

**Criterios de aceptación** Ocultar elementos completados Eliminar

100%

- Al dar clic en un tipo de teléfono se debe de mostrar las siguiente opciones: Editar y Eliminar.
- Al dar clic en el botón "Editar", este debe de direccionar a ala vista de Editar tipo de teléfono.
- Botón "Eliminar" para borrar el tipo de teléfono seleccionado.
- Al dar clic en el botón Eliminar, este debe de mostrar una ventana emergente de confirmación. Si se da aceptar el sistema debe de eliminar el tipo de teléfono del sistema.

Añada un elemento

**AÑADIR A LA TARJETA**

- Miembros
- Etiquetas
- Checklist
- Vencimiento
- Adjunto
- Portada

**POWER-UPS**

Conseguir Power-Ups

Consiga Power-Ups ilimitados y mucho más.

- Actualizar el equipo

**ACCIONES**

- Mover

Figura 7.1.70

Autor: Auditoría propia, herramienta usada: Trello.

- HU 71

**HU - 71. Editar o actualizar la información de un tipo de teléfono.**  
en la lista [Sprint 7](#)

**Descripción** [Editar](#)

Yo como administrador del sistema requiero que la aplicación permita actualizar la información de un tipo de teléfono seleccionado, para poder corregir información o actualizarla.

**Criterios de aceptación** [Ocultar elementos completados](#) [Eliminar](#)

100%

- Se debe de mostrar la siguiente información: Tipo de teléfono.
- Si se actualiza el campo "Tipo de teléfono.", solo aceptará una descripción con al menos 2 caracteres.
- El campo "Tipo de teléfono." no puede estar vacío(Nulo).
- Botón para cancelar el proceso.
- Al dar clic en cancelar, este debe de direccionar a la vista del ver tipos de teléfonos.
- Botón para actualizar lo datos.
- Los datos actualizados deben de versen reflejados al ver de nuevo la información del tipo de teléfono seleccionado.

[Añada un elemento](#)

**AÑADIR A LA TARJETA**

- [Miembros](#)
- [Etiquetas](#)
- [Checklist](#)
- [Vencimiento](#)
- [Adjunto](#)
- [Portada](#)

**POWER-UPS**

[Conseguir Power-Ups](#)

Consiga Power-Ups ilimitados y mucho más.

[Actualizar el equipo](#)

**ACCIONES**

- [→ Mover](#)
- [Copiar](#)
- [Compartir](#)

Figura 7.1.71

Autor: Auditoría propia, herramienta usada: Trello.

- HU 72

The screenshot shows a Trello card titled "HU - 72. Vista para ver los Roles o cargos de la sede." with a sub-header "en la lista [Sprint 7](#)". The card is organized into sections:

- Descripción**: Includes an "Editar" button and the text: "Yo como administrador del sistema requiero que la aplicación permita observar los Roles o cargos que tiene el sistema, para poder ver los roles que se usan asignar a un Empleado."
- Criterios de aceptación**: Includes buttons for "Ocultar elementos completados" and "Eliminar". A progress bar shows 100% completion.
- Checklist**: Contains two items, both checked:
  - ✓ El sistema debe de mostrar la siguiente información: Rol
  - ✓ Botón para regresar a la vista del módulo de talento humano.

On the right side, there is a sidebar with the heading "AÑADIR A LA TARJETA" and several options: "Miembros", "Etiquetas", "Checklist", "Vencimiento", "Adjunto", and "Portada". Below this is a "POWER-UPS" section with a partially visible option "Compartir Datos Usar".

Figura 7.1.72

Autor: Auditoría propia, herramienta usada: Trello.

- HU 73

**HU - 73. Ingresar una nueva fórmula.**  
en la lista [Sprint 7](#)

**Descripción** Editar

Yo como administrador del sistema requiero que la aplicación permita ingresar una nueva formula, para guardarla y hacer uso de ella como receta para hacer los productos.

**Criterios de aceptación** Ocultar elementos completados Eliminar

100%

- ✓ La aplicación debe permitir ingresar la siguiente información: Producto, Nombre, Cantidad estimada, Descripción 1, Descripción 2.
- ✓ En el campo "Producto", el sistema debe de mostrar los productos disponibles y permitir elegir una de ellos.
- ✓ El campo "Nombre" solo aceptará un nombre con al menos 4 caracteres.
- ✓ El campo "Cantidad estimada" solo aceptará números o decimales.
- ✓ El campo "Descripción 1", solo aceptará una Descripción con al menos 6 caracteres.
- ✓ El campo "Descripción 2", solo aceptará una Descripción con al menos 6 caracteres.
- ✓ Todos los campos son obligatorios.
- ✓ Botón para crear o guardar la fórmula.
- ✓ Botón para cancelar el ingreso de la fórmula.
- ✓ Al dar clic en el botón cancelar, este debe de direccionar a la vista de ver fórmulas.

Añada un elemento

**AÑADIR A LA TARJETA**

- Miembros
- Etiquetas
- Checklist
- Vencimiento
- Adjunto
- Portada

**POWER-UPS**

Conseguir Power-Ups

Consiga Power-Ups ilimitados y mucho más.

Actualizar el equipo

**ACCIONES**

- Mover
- Copiar
- Convertir en plant...
- Seguir
- Archivar
- Compartir

Figura 7.1.73

Autor: Auditoría propia, herramienta usada: Trello.

- HU 74

**HU - 74. Vista para ver las fórmulas del sistema.**  
en la lista [Sprint 7](#)

**Descripción** Editar

Yo como administrador del sistema requiero que la aplicación permita ver las fórmulas que tiene la sede, Para poder elegir y observar la receta de la fórmula.

**Criterios de aceptación** Ocultar elementos completados Eliminar

100%

- El sistema debe de mostrar la siguiente información por cada fórmula: Fórmula, Cantidad de producto estimada y Costo.
- Filtro para buscar una fórmula por su nombre.
- Botón para ingresar una nueva fórmula.
- Botón para regresar al módulo de producción.

Añada un elemento

**AÑADIR A LA TARJETA**

- Miembros
- Etiquetas
- Checklist
- Vencimiento
- Adjunto
- Portada

**POWER-UPS**

Conseguir Power-Ups

Consiga Power-Ups ilimitados y mucho más.

Figura 7.1.74

Autor: Auditoría propia, herramienta usada: Trello.

- HU 75

**HU - 75 Calcular el costo de una fórmula.**  
en la lista [Sprint 7](#)

**Descripción** Editar

Yo como administrador del sistema requiero que la aplicación calcule el costo de una fórmula, para saber a que precio es rentable vender el producto final que sale de dicha fórmula.

Cuando se crea una fórmula esta tiene un costo de 0 pesos, entonces cada vez que se le agregue un nuevo insumo a la fórmula se debe de calcular el costo a partir del valor de la unidad del insumo que se tenga en el inventario de materias primas.

Tener en cuenta que si se actualiza el valor de unidad de una materia prima desde el inventario, y esta se está usando como insumo de una fórmula, se tiene que ver reflejado el cambio en el costo de la fórmula.

También, si se le quita algún ingrediente a la fórmula, el costo de la fórmula debe de quedar actualizado y correctamente.

- La calculo se debe de realizar de la siguiente manera:

Costo = valor formula +(Valor unidad \* cantidad de ingrediente )

**Criterios de aceptación** Ocultar elementos completados Eliminar

100%

- Cada fórmula debe de tener su valor de costo calculado.
- El calculo debe de ser correcto de acuerdo a la expresión matemática de la descripción.
- Si se actualiza el valor de la unidad en el inventario, el costo de la formula también debe de actualizar.
- Si se quita un ingrediente, el costo de la formula debe de actualizarse.

Añada un elemento

**AÑADIR A LA TARJETA**

- Miembros
- Etiquetas
- Checklist
- Vencimiento
- Adjunto
- Portada

**POWER-UPS**

Conseguir Power-Ups

Consiga Power-Ups ilimitados y mucho más.

[Actualizar el equipo](#)

**ACCIONES**

- Mover
- Copiar
- Convertir en plant...
- Seguir
- Archivar
- Compartir

Figura 7.1.75

Autor: Auditoría propia, herramienta usada: Trello.

- HU 76

**HU -76. Implementar opciones al seleccionar una fórmula.**  
en la lista [Sprint 7](#)

**Descripción** Editar

Yo como administrador del sistema requiero que la aplicación permita seleccionar una fórmula para:

- Ver los ingredientes de la fórmula seleccionada.
- Actualizar la información de la fórmula seleccionada.
- Eliminar la fórmula seleccionada.
- Ver la información detallada de la fórmula seleccionada.

Nota: El botón de eliminar estará activó siempre y cuando la fórmula no tenga ingredientes . Si la fórmula tiene ingredientes el botón de eliminar desaparecerá.

**Criterios de aceptación** Ocultar elementos completados Eliminar

100%

- ✓ Al seleccionar una fórmula, se debe de mostrar las siguientes opciones: Ver ingredientes, Editar, Eliminar y Ver información.
- ✓ Botón para ver los ingredientes de la fórmula.
- ✓ Al dar clic en el botón "Ver ingredientes", debe de direccionar a la vista de ver ingredientes.
- ✓ Botón para actualizar la información de una fórmula.
- ✓ Al dar clic en el botón de "Editar", este debe de direccionar a la vista de Editar fórmula.
- ✓ Botón para eliminar la fórmula.
- ✓ Al dar clic en el botón Eliminar, este debe de mostrar una ventana emergente de confirmación. Si se da aceptar el sistema debe de eliminar la fórmula del sistema.
- ✓ Si la fórmula tiene un ingrediente ingresado , el botón de eliminar no se debe mostrar.
- ✓ Botón para ver información detallada de la fórmula.
- ✓ Al dar clic en el botón de "ver información", este debe de direccionar a la vista de ver información fórmula.

Añada un elemento

**AÑADIR A LA TARJETA**

- Miembros
- Etiquetas
- Checklist
- Vencimiento
- Adjunto
- Portada

**POWER-UPS**

Conseguir Power-Ups

Consiga Power-Ups ilimitados y mucho más.

- Actualizar el equipo

**ACCIONES**

- Mover
- Copiar
- Convertir en plant...
- Seguir
- Archivar
- Compartir

Figura 7.1.76

Autor: Auditoría propia, herramienta usada: Trello.

- HU 77

**HU - 77. Editar o actualizar la información de una fórmula.**  
en la lista [Sprint 7](#)

**Descripción** Editar

Yo cómo administrador del sistema requiero que la aplicación permita actualizar la información de una fórmula, para corregir datos mal ingresados o para agregar más información a la fórmula.

**Criterios de aceptación** Ocultar elementos completados Eliminar

100%

- ✓ La aplicación debe permitir ingresar la siguiente información: Producto, Nombre, Cantidad estimada, Descripción 1, Descripción 2.
- ✓ Si se actualiza el campo "Producto", el sistema debe de mostrar los productos disponibles y permitir elegir una de ellos.
- ✓ Si se actualiza el campo "Nombre" solo aceptará un nombre con al menos 4 caracteres.
- ✓ Si se actualiza el campo "Cantidad estimada" solo aceptará números o decimales.
- ✓ Si se actualiza el campo "Descripción 1", solo aceptará una Descripción con al menos 6 caracteres.
- ✓ Si se actualiza el campo "Descripción 2", solo aceptará una Descripción con al menos 6 caracteres.
- ✓ Ningún campo puede estar vacío (Nulo).
- ✓ Botón para actualizar la información de la fórmula.
- ✓ Botón para cancelar el proceso.
- ✓ Al dar clic en el botón cancelar, este debe de direccionar a la vista de ver fórmulas.

Añada un elemento

**AÑADIR A LA TARJETA**

- Miembros
- Etiquetas
- Checklist
- Vencimiento
- Adjunto
- Portada

**POWER-UPS**

Conseguir Power-Ups

Consiga Power-Ups ilimitados y mucho más.

- Actualizar el equipo

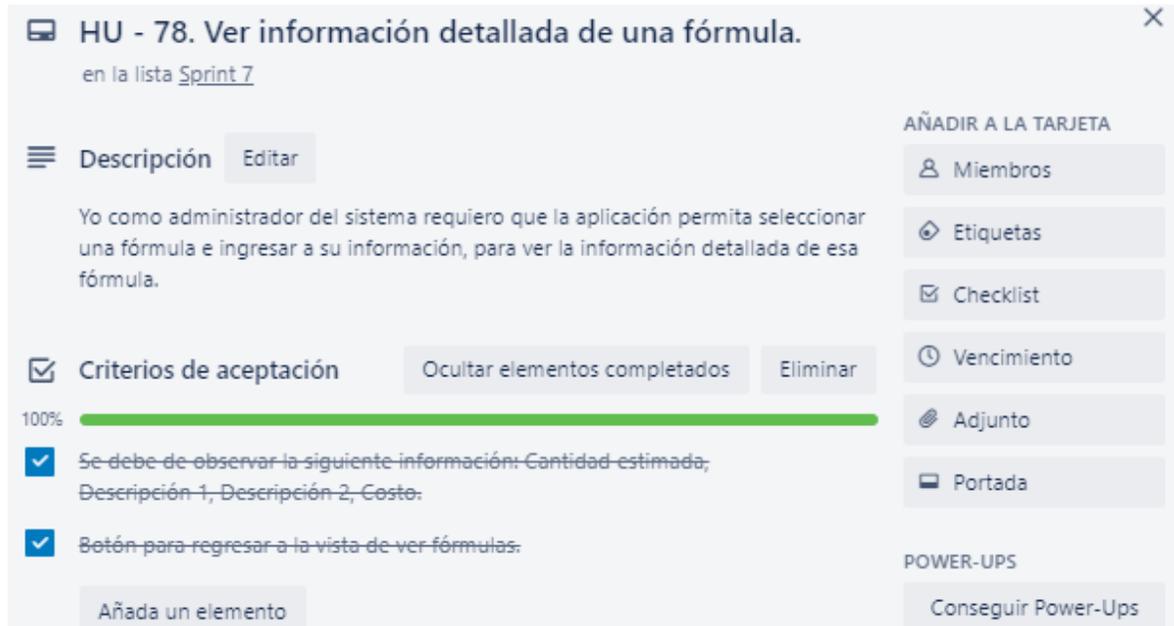
**ACCIONES**

- Mover
- Copiar
- Convertir en plant...
- Seguir
- Archivar
- Compartir

Figura 7.1.77

Autor: Auditoría propia, herramienta usada: Trello.

- HU 78



The image shows a Trello card titled "HU - 78. Ver información detallada de una fórmula." located in the "Sprint 7" list. The card is divided into several sections:

- Header:** "HU - 78. Ver información detallada de una fórmula." with a close button (X) in the top right corner. Below the title, it says "en la lista [Sprint 7](#)".
- Description:** A section with a "Descripción" label and an "Editar" button. The text reads: "Yo como administrador del sistema requiero que la aplicación permita seleccionar una fórmula e ingresar a su información, para ver la información detallada de esa fórmula."
- Acceptance Criteria:** A section with a "Criterios de aceptación" label, an "Ocultar elementos completados" button, and an "Eliminar" button. It shows a 100% progress bar (green) and two checked items:
  - ✓ Se debe de observar la siguiente información: Cantidad estimada, Descripción 1, Descripción 2, Costo.
  - ✓ Botón para regresar a la vista de ver fórmulas.Below this section is an "Añada un elemento" button.
- Right Sidebar (Add-ons):** A section titled "AÑADIR A LA TARJETA" with buttons for "Miembros", "Etiquetas", "Checklist", "Vencimiento", "Adjunto", and "Portada". Below this is a "POWER-UPS" section with a "Conseguir Power-Ups" button.

Figura 7.1.78

Autor: Auditoría propia, herramienta usada: Trello.

- HU 79

The image shows a Trello card titled "HU - 79. Ver ingredientes de una fórmula." located in the "Sprint 7" list. The card content includes a description, acceptance criteria, and a progress bar. On the right side, there are several utility buttons for adding items to the card and power-ups.

**Card Title:** HU - 79. Ver ingredientes de una fórmula.  
**Location:** en la lista [Sprint 7](#)

**Section: Descripción** Editar

yo como administrador del sistema requiero que la aplicación permita ver los ingredientes que tiene una formula con su respectiva cantidad, para poder preparar la fórmula.

**Section: Criterios de aceptación** Ocultar elementos completados Eliminar

100% Progress bar

- El sistema debe de mostrar la siguiente información: Ingrediente; Cantidad;
- La cantidad se debe demostrar con su respectiva unidad de medida.
- Botón para agregar un nuevo ingrediente a la fórmula.
- Botón para regresar a la vista de fórmulas.

Añada un elemento

**Utility Buttons (AÑADIR A LA TARJETA):**

- Miembros
- Etiquetas
- Checklist
- Vencimiento
- Adjunto
- Portada

**POWER-UPS**

- Conseguir Power-Ups

Consiga Power-Ups ilimitados y mucho más.  
Actualizar el equipo

Figura 7.1.79

Autor: Auditoría propia, herramienta usada: Trello.

- HU 80

**HU - 80. Ingresar ingredientes a una fórmula.**  
en la lista [Sprint 7](#)

**Descripción** Editar

Yo como administrador requiero que la aplicación permita ingresar diferentes ingredientes a una fórmula ya registrada, para poder construir la fórmula con lo que necesita.

Se requiere que el sistema liste o muestre los insumos disponibles del inventario, para poder elegir uno de ellos y agregarlo a la fórmula.

Cuando se seleccione un insumo para agregarlo a la fórmula, se debe de abrir una ventana que solicite la cantidad que necesita la fórmula de ese insumo.

**Criterios de aceptación** Ocultar elementos completados Eliminar

100%

- ✓ El sistema debe de listar los insumos disponible con la siguiente información: El nombre del insumo y la unidad de medida del insumo.
- ✓ Filtro para buscar un insumo por su nombre.
- ✓ Al dar clic en insumo se debe abrir una nueva ventana con las siguientes características: Campo solicitando la cantidad del insumo; Botón para agregar la cantidad, botón para cancelar.
- ✓ El campo de cantidad de insumo, sólo aceptará números y decimales.
- ✓ El campo de cantidad de insumo es obligatorio.
- ✓ Cuando se de clic en el botón de agregar cantidad, se debe de agregar el insumo a la lista de ingredientes.
- ✓ Al dar clic en el botón de agregar cantidad, este debe de direccionar a la vista de ver ingredientes de la fórmula. Y se debe de observar el nuevo insumo agregado.
- ✓ Botón agregar insumo.
- ✓ Botón para cancelar el proceso.

Añada un elemento

**AÑADIR A LA TARJETA**

- Miembros
- Etiquetas
- Checklist
- Vencimiento
- Adjunto
- Portada

**POWER-UPS**

Conseguir Power-Ups

Consiga Power-Ups ilimitados y mucho más.

Actualizar el equipo

**ACCIONES**

- Mover
- Copiar
- Convertir en plant...
- Seguir
- Archivar
- Compartir

Figura 7.1.80

Autor: Auditoría propia, herramienta usada: Trello.

- HU 81

**HU - 81. Implementar opciones al seleccionar el ingrediente de una fórmula.**  
en la lista [Sprint 7](#)

**Descripción** Editar

Yo como administrador del sistema requiero que la aplicación permita seleccionar un ingrediente para:

- Editar o actualizar la información del ingrediente seleccionado.
- Eliminar el ingrediente seleccionado.

**✓ Criterios de aceptación** Ocultar elementos completados Eliminar

100%

- ✓ Al dar clic en un ingrediente de la fórmula se debe de mostrar las siguientes opciones: Editar y Eliminar.
- ✓ Al dar clic en el botón "Editar", este debe de direccionar a ala vista de Editar Ingrediente.
- ✓ Botón "Eliminar" para borrar el ingrediente seleccionado.
- ✓ Al dar clic en el botón Eliminar, este debe de mostrar una ventana emergente de confirmación. Si se da aceptar el sistema debe de eliminar el ingrediente del sistema.

Añada un elemento

**AÑADIR A LA TARJETA**

- Miembros
- Etiquetas
- Checklist
- Vencimiento
- Adjunto
- Portada

**POWER-UPS**

Conseguir Power-Ups

Consiga Power-Ups ilimitados y mucho más.

Actualizar el equipo

**ACCIONES**

- Mover

Figura 7.1.81

Autor: Auditoría propia, herramienta usada: Trello.

- HU 82

**HU - 82. Editar o actualizar la información del ingrediente de una fórmula.**  
en la lista [Sprint 7](#)

**Descripción** Editar

Yo como administrador del sistema requiero que la aplicación permita actualizar la información de un ingrediente seleccionado, para poder corregir información o actualizarla.

**✓ Criterios de aceptación** Ocultar elementos completados Eliminar

100%

- ✓ Se debe de mostrar la siguiente información: La cantidad del ingrediente.
- ✓ Si se actualiza el campo "La cantidad del ingrediente", solo aceptará números y decimales.
- ✓ El campo "La cantidad del ingrediente" no puede estar vacío (Nulo).
- ✓ Botón para actualizar los datos.
- ✓ Los datos actualizados deben de verse reflejados al ver de nuevo la información del insumo.
- ✓ Botón para cancelar el proceso.
- ✓ Al dar clic en cancelar, este debe de direccionar a la vista del ver ingredientes de la fórmula.

Añada un elemento

**AÑADIR A LA TARJETA**

- Miembros
- Etiquetas
- Checklist
- Vencimiento
- Adjunto
- Portada

**POWER-UPS**

Conseguir Power-Ups

Consiga Power-Ups ilimitados y mucho más.

Actualizar el equipo

**ACCIONES**

- Mover
- Copiar
- Convertir en plant...

Figura 7.1.82

Autor: Auditoría propia, herramienta usada: Trello.

## Sprint 8

- HU 83

**HU - 83. Ingresar un producto al sistema.**  
en la lista [Sprint 8](#)

**Descripción** Editar

Yo como administrador del sistema requiero que la aplicación permita ingresar un nuevo producto, para poder hacer uso de este en la creación de una fórmula y así mismo, poder generar la producción de ese producto.

**Criterios de aceptación** Ocultar elementos completados Eliminar

100%

- ✓ El sistema debe de permitir ingresar la siguiente información: Nombre, Descripción:
- ✓ El campo "Nombre", solo aceptará un Nombre con al menos 4 caracteres:
- ✓ El campo "Descripción", solo aceptará una descripción con al menos 4 caracteres:
- ✓ Todos los campos son obligatorios:
- ✓ Botón para agregar el producto al sistema:
- ✓ Botón para cancelar el proceso:
- ✓ Al dar clic en el botón cancelar, este debe de direccionar a la vista de ver productos:

Añada un elemento

**AÑADIR A LA TARJETA**

- Miembros
- Etiquetas
- Checklist
- Vencimiento
- Adjunto
- Portada

**POWER-UPS**

Conseguir Power-Ups

Consiga Power-Ups ilimitados y mucho más.

Actualizar el equipo

**ACCIONES**

- Mover
- Copiar
- Convertir en plant...

Figura 7.1.83

Autor: Auditoría propia, herramienta usada: Trello.

- HU 84

**HU - 84. Ver los productos que tiene el sistema.**  
en la lista [Sprint 8](#)

**Descripción** Editar

Yo como administrador del sistema requiero que la aplicación permita observar los productos que se han ingresado al sistema, para saber que tipo de producción se realizará y se llevará a la venta.

**Criterios de aceptación** Ocultar elementos completados Eliminar

100%

- El sistema debe de mostrar la siguiente información: Nombre, Descripción.
- Filtro para buscar un producto por su nombre.
- Botón para ingresar un nuevo producto.
- Botón para regresar a la vista del módulo de producción.

Añada un elemento

**AÑADIR A LA TARJETA**

- Miembros
- Etiquetas
- Checklist
- Vencimiento
- Adjunto
- Portada

**POWER-UPS**

Conseguir Power-Ups

Consiga Power-Ups ilimitados y mucho más.

Actualizar el equipo

Figura 7.1.84

Autor: Auditoría propia, herramienta usada: Trello.

- HU 85

**HU -85. Implementar opciones al seleccionar un producto.**  
en la lista [Sprint 8](#)

**Descripción** Editar

Yo como administrador del sistema requiero que la aplicación permita seleccionar un producto para:

- Editar o actualizar la información del producto seleccionado.
- Eliminar el producto seleccionado.

Nota: El botón de eliminar estará activo siempre y cuando se halla usado para crear una fórmula. Si se usó el botón debe de desaparecer.-

**Criterios de aceptación** Ocultar elementos completados Eliminar

100%

- ✓ Al dar clic en un producto se debe de mostrar las siguientes opciones: Editar y Eliminar.
- ✓ Al dar clic en el botón "Editar", este debe de direccionar a la vista de Editar Producto.
- ✓ Botón "Eliminar" para borrar el producto seleccionado.
- ✓ Al dar clic en el botón Eliminar, este debe de mostrar una ventana emergente de confirmación. Si se da aceptar el sistema debe de eliminar el producto del sistema.
- ✓ Si el producto fue usado para crear una fórmula, el botón de eliminar no se debe mostrar.
- ✓ Botón "Editar" para ir a actualizar la información de un producto.

Añada un elemento

**AÑADIR A LA TARJETA**

- Miembros
- Etiquetas
- Checklist
- Vencimiento
- Adjunto
- Portada

**POWER-UPS**

Conseguir Power-Ups

Consiga Power-Ups ilimitados y mucho más.

Actualizar el equipo

**ACCIONES**

- Mover
- Copiar
- Convertir en plant...
- Seguir
- Archivar

Figura 7.1.85

Autor: Auditoría propia, herramienta usada: Trello.

- HU 86

**HU - 86. Editar o actualizar la información de un producto.**  
en la lista [Sprint 8](#)

**Descripción** Editar

Yo como administrador del sistema requiero que la aplicación permita actualizar la información de un tipo producto seleccionado, para poder corregir información o actualizarla.

**Criterios de aceptación** Ocultar elementos completados Eliminar

100%

- ✓ Se debe de mostrar la siguiente información: Nombre, Descripción.
- ✓ Si se actualiza el campo "Descripción", solo aceptará una descripción con al menos 4 caracteres.
- ✓ Si se actualiza el campo "Nombre", solo aceptará un nombre con al menos 4 caracteres.
- ✓ Ningún campo puede estar vacío(Nulo).
- ✓ Botón para actualizar lo datos.
- ✓ Los datos actualizados deben de verse reflejados al ver de nuevo la información del producto seleccionado.
- ✓ Botón para cancelar el proceso.
- ✓ Al dar clic en cancelar, este debe de direccionar a la vista del ver productos.

Añada un elemento

**AÑADIR A LA TARJETA**

- Miembros
- Etiquetas
- Checklist
- Vencimiento
- Adjunto
- Portada

**POWER-UPS**

Conseguir Power-Ups

Consiga Power-Ups ilimitados y mucho más.

Actualizar el equipo

**ACCIONES**

- Mover
- Copiar
- Convertir en plant...
- Seguir

Figura 7.1.86

Autor: Auditoría propia, herramienta usada: Trello.

- HU 87

**HU - 87. Vista de fabricación de mojes.**  
en la lista [Sprint 8](#)

**Descripción** Editar

Yo como administrador del sistema requiero que la aplicación liste las formulas que el sistema tiene actualmente y permita seleccionar una de ellas para iniciar la fabricación de un nuevo moje.

**Criterios de aceptación** Ocultar elementos completados Eliminar

100%

- El sistema debe de mostrar la siguiente información por cada fórmula: El nombre de la fórmula.
- Filtro para buscar una fórmula por su nombre.
- Al dar clic en una fórmula, este debe de direccionar a la vista de ingredientes de la fórmula seleccionada.

Añada un elemento

**AÑADIR A LA TARJETA**

- Miembros
- Etiquetas
- Checklist
- Vencimiento
- Adjunto
- Portada

**POWER-UPS**

Conseguir Power-Ups

Consiga Power-Ups ilimitados y mucho más.

Actualizar el equipo

Figura 7.1.87

Autor: Auditoría propia, herramienta usada: Trello.

- HU 88

**HU - 88. Vista de ingredientes para la fabricación de un moje.**  
en la lista [Sprint 8](#)

**Descripción** Editar

Yo como administrador del sistema requiero que la aplicación tenga una vista donde se pueda observar los ingredientes que lleva la formula con su respectiva descripción, para que el panadero pueda seguir paso a paso las instrucciones de la fórmulas y construir un moje.

Sólo se podrá fabricar un moje si existe la cantidad de insumos suficientes en el inventario, de lo contrario se debe de desaparecer el botón para crear el moje.

Se debe de pintar de un color rojo los ingredientes que no tengan la suficiente cantidad en el inventario para crear el moje, para poder identificarlos fácilmente.

La formula se selecciona en la vista de fabricación de mojes.

**Criterios de aceptación** Ocultar elementos completados Eliminar

100%

- El sistema debe de mostrar la siguiente información por cada ingrediente: El nombre del ingrediente, Cantidad, Inventario y la descripción de la fórmula.
- Botón para crear un moje de la fórmula.
- Botón para regresar a la vista de fabricación de mojes.
- Se deben de pintar de color rojo los ingredientes que tienen cantidad insuficiente a la cantidad que necesita la fórmula para fabricar el moje.
- En caso de no tener la cantidad suficiente de ingrediente, el botón para fabricar el moje debe de desaparecer.
- Si el botón desaparece se debe de mostrar un mensaje con la siguiente información: **NO SE PUEDE FABRICAR EL MOJE POR FALTA DE INSUMOS EN EL INVENTARIO**

Añada un elemento

**AÑADIR A LA TARJETA**

- Miembros
- Etiquetas
- Checklist
- Vencimiento
- Adjunto
- Portada

**POWER-UPS**

Conseguir Power-Ups

Consiga Power-Ups ilimitados y mucho más.

Actualizar el equipo

**ACCIONES**

- Mover
- Copiar
- Convertir en plant...
- Seguir
- Archivar
- Compartir

Figura 7.1.88

Autor: Auditoría propia, herramienta usada: Trello.

- HU 89

**HU - 89. Crear moje de una fórmula.**  
en la lista [Sprint 8](#)

**Descripción** Editar

Yo como administrador del sistema requiero que la aplicación permita crear un moje de una fórmula para que el panadero pueda fabricar el moje y generar la producción de la formula de dicho moje.

Cuando se le de clic en crear un moje, se debe de descontar la cantidad de ingrediente que utilizó el moje del inventario de materias primas. Se debe de mostrar un mensaje de confirmación para crear el moje.

**Criterios de aceptación** Ocultar elementos completados Eliminar

100%

- ✓ El sistema debe de permitir ingresar la siguiente información: Fecha de creación del moje, Peso del moje, Observaciones.
- ✓ El campo "Fecha de creación" sólo aceptará la fecha en formato ss/mm/aaaa
- ✓ El campo "Peso moje", solo aceptará números o decimales.
- ✓ El campo "Observación", solo aceptará una observación que tenga cómo mínimo 4 caracteres.
- ✓ Todos los campos son obligatorios.
- ✓ Botón para crear el moje.
- ✓ Botón para cancelar el proceso.
- ✓ Al dar clic en el botón cancelar este debe de direccionar a la vista de "ingredientes para la fabricación de un moje."
- ✓ Al crear el moje se debe de evidenciar que los ingredientes del moje se redujeron en el inventario de materias primas.
- ✓ Mensaje de confirmación para crear el moje.

**AÑADIR A LA TARJETA**

- Miembros
- Etiquetas
- Checklist
- Vencimiento
- Adjunto
- Portada

**POWER-UPS**

Conseguir Power-Ups

Consiga Power-Ups ilimitados y mucho más.

Actualizar el equipo

**ACCIONES**

- Mover
- Copiar
- Convertir en plant...
- Seguir
- Archivar
- Compartir

Figura 7.1.89

Autor: Auditoría propia, herramienta usada: Trello.

- HU 90

**HU -90. Ver inventarios de mojes.**  
en la lista [Sprint 8](#)

**Descripción** Editar

Yo como administrador del sistema requiero que la aplicación permita ver los mojes que se han fabricado, ordenados por su fecha de fabricación, para saber que mojes fueron fabricados y finalizados como un producto final.

El sistema debe de mostrar con un color verde el moje que aún no se halla terminado, es decir, el moje que aún falte por generar el producto terminado.

**Criterios de aceptación** Ocultar elementos completados Eliminar

100%

- El sistema debe de mostrar la siguiente información por cada moje fabricado: El nombre de la fórmula del moje, Fecha de creación, peso en libras del moje.
- Filtro para buscar un moje por su nombre de fórmula.
- Filtro para buscar un moje por fecha de fabricación.
- Se debe de pintar de color verde el moje que no se halla terminado, es decir, aquel que no halla registrado el producto terminado.
- Botón para regresar al módulo de producción.

Añada un elemento

**AÑADIR A LA TARJETA**

- Miembros
- Etiquetas
- Checklist
- Vencimiento
- Adjunto
- Portada

**POWER-UPS**

Conseguir Power-Ups

Consiga Power-Ups ilimitados y mucho más.

Actualizar el equipo

**ACCIONES**

- Mover
- Copiar

Figura 7.1.90

Autor: Auditoría propia, herramienta usada: Trello.

- HU 91

**HU - 91. Implementar opciones al seleccionar un moje del inventario.**  
en la lista [Sprint 8](#)

**Descripción** Editar

Yo como administrador del sistema requiero que la aplicación permita seleccionar del inventario de mojes, un moje para:

- Ver la información detallada del moje fabricado.
- Registrar el producto terminado del moje.

Sólo debe de aparecer la opción del producto terminado, si el moje seleccionado no se le ha registrado el producto terminado. Si el moje ya se le registró el producto terminado, no debe de aparecer el botón para registrarlo.

**Criterios de aceptación** Ocultar elementos completados Eliminar

100%

- ✓ Al dar clic en un producto se debe de mostrar las siguientes opciones: Registrar producto terminado y Ver información.
- ✓ Al dar clic en el botón "Registrar producto terminado", este debe de direccionar a la vista de Registrar producto terminado.
- ✓ Al dar clic en el botón "Ver información", este debe de direccionar a la vista de Ver información detallada del moje fabricado.
- ✓ Sólo debe de aparecer la opción del producto terminado, si el moje seleccionado no se le ha registrado el producto terminado
- ✓ Si el moje ya se le registró el producto terminado, no debe de aparecer el botón para registrarlo.

Añada un elemento

**AÑADIR A LA TARJETA**

- Miembros
- Etiquetas
- Checklist
- Vencimiento
- Adjunto
- Portada

**POWER-UPS**

Conseguir Power-Ups

Consiga Power-Ups ilimitados, mucho más.

- Actualizar el equipo

**ACCIONES**

- Mover
- Copiar
- Convertir en plant...
- Seguir

Figura 7.1.91

Autor: Auditoría propia, herramienta usada: Trello.

- HU 92

**HU - 92. Registrar un producto terminado.**  
en la lista [Sprint 8](#)

**Descripción** Editar

Yo como administrador del sistema requiero que la aplicación permita registrar un producto terminado de un moje, para saber que productos finales salieron del moje.

**Criterios de aceptación** Ocultar elementos completados Eliminar

100%

- ✓ El sistema debe de permitir ingresar la siguiente información: Cantidad de productos que salieron del moje y observación.
- ✓ El campo "Cantidad" sólo aceptará número y decimales.
- ✓ El campo "Observación", solo aceptará una observación que tenga cómo mínimo 4 caracteres.
- ✓ Todos los campos son obligatorios.
- ✓ Botón para crear el producto terminado.
- ✓ Botón para cancelar el proceso.
- ✓ Al dar clic en el botón cancelar este debe de direccionar a la vista de "Inventarios de mojes."
- ✓ Al dar clic en el botón crear la aplicación debe de ingresar toda la información del producto terminado al sistema.
- ✓ El producto terminado registrado debe de aparecer en el inventario de productos terminados.

Añada un elemento

**AÑADIR A LA TARJETA**

- Miembros
- Etiquetas
- Checklist
- Vencimiento
- Adjunto
- Portada

**POWER-UPS**

Conseguir Power-Ups

Consiga Power-Ups ilimitado mucho más.

Actualizar el equipo

**ACCIONES**

- Mover
- Copiar
- Convertir en plant...
- Seguir
- Archivar

Figura 7.1.92

Autor: Auditoría propia, herramienta usada: Trello.

- HU 93

The image shows a Trello card titled "HU - 93. Ver información detallada de un moje," located in the "Sprint 8" list. The card is organized into several sections:

- Header:** Title "HU - 93. Ver información detallada de un moje," and location "en la lista Sprint 8".
- Description:** A text block stating: "Yo como administrador del sistema requiero que la aplicación permita seleccionar un moje fabricado e ingresar a su información, para ver la información detallada de ese moje." It includes an "Editar" button.
- Acceptance Criteria:** A section titled "Criterios de aceptación" with a 100% completion bar. It contains two checked items:
  - ✓ Se debe de observar la siguiente información: Fórmula, Fecha de creación, peso, Producto, Cantidad de producto estimado, Cantidad de producto real, Observación.
  - ✓ Botón para regresar a la vista de inventario de mojes.Buttons for "Ocultar elementos completados" and "Eliminar" are present.
- Power-Ups:** A sidebar on the right titled "AÑADIR A LA TARJETA" with options: "Miembros", "Etiquetas", "Checklist", "Vencimiento", "Adjunto", and "Portada". Below it is a "POWER-UPS" section with a "Conseguir Power-Ups" button.

Figura 7.1.93

Autor: Auditoría propia, herramienta usada: Trello.

- HU 94

**HU - 94. Ver productos terminados.**  
en la lista [Sprint 8](#)

**Descripción** Editar

Yo como administrador del sistema requiero que la aplicación permita observar los productos terminados, para:

- Conocer la producción que se ha realizado en la sede.
- Conocer que productos fueron realizados y de que fórmula salieron.
- Conocer que cantidad salió del producto con base a la que se estima en la fórmula.
- Visualizar si el producto generó pérdida en términos de porcentajes..

El sistema debe de pintar de color rojo los productos finales que generaron pérdida. El porcentaje de pérdida esta definido como del 4% para arriba.

Este porcentaje de pérdida se debe de calcular de la siguiente manera:

- $\text{Porcentaje} = 100 - ((\text{cantidad estimada} * 100) / \text{cantidad real})$

**Criterios de aceptación** Ocultar elementos completados Eliminar

100%

- El sistema debe de mostrar la siguiente información por cada producto final: El nombre de Producto, Fórmula, Cantidad real, Cantidad estimada, % de pérdidas.
- Filtro para buscar un producto terminado por el nombre del producto.
- Filtro para buscar un producto terminado por el nombre de su fórmula.
- Botón para regresar al módulo de producción.
- Debe de aparecer en color rojo el producto final que halla generado pérdida.
- Sólo se debe de pintar los productos finales de color rojo que cumplan con el % de pérdida definido en la descripción de la HU.

Añade un elemento

**AÑADIR A LA TARJETA**

- Miembros
- Etiquetas
- Checklist
- Vencimiento
- Adjunto
- Portada

**POWER-UPS**

Conseguir Power-Ups

Consiga Power-Ups ilimitados mucho más.

- Actualizar el equipo

**ACCIONES**

- Mover
- Copiar
- Convertir en plant...
- Seguir
- Archivar
- Compartir

Figura 7.1.94

Autor: Auditoría propia, herramienta usada: Trello.

- HU 95

The image shows a Trello card titled "HU - 95. Implementar opciones al seleccionar un producto terminado." located in the "Sprint 8" list. The card is organized into several sections:

- Descripción:** A section with a "Descripción" header and an "Editar" button. The text reads: "Yo como administrador del sistema requiero que la aplicación permita seleccionar un producto terminado para:" followed by a bullet point: "• Ver la información detallada del producto terminado."
- Criterios de aceptación:** A section with a "Criterios de aceptación" header, an "Ocultar elementos completados" button, and an "Eliminar" button. It features a 100% progress bar and two checked items:
  - ✓ Al dar clic en un producto final se debe de mostrar las siguientes opciones: Ver información.
  - ✓ Al dar clic en el botón "Ver información", este debe de direccionar a la vista de Ver información detallada del producto final.
- Añadir a la tarjeta:** A vertical sidebar on the right with buttons for "Miembros", "Etiquetas", "Checklist", "Vencimiento", "Adjunto", and "Portada".
- Power-Ups:** A section at the bottom right with a "Conseguir Power-Ups" button and the text "Consiga Power-Ups ilimitados y mucho más."

Figura 7.1.95

Autor: Auditoría propia, herramienta usada: Trello.

## Sprint 9

- HU 96

**HU - 96. Vista para fabricar un producto terminado.**  
en la lista [Sprint 9](#)

**Descripción** Editar

Yo como administrador del sistema requiero que la aplicación permita ver los mojes que aún no se han terminado, es decir, que aún no se han generado como producto terminado, para poder elegirlo y generarlo como un producto final.

**Criterios de aceptación** Ocultar elementos completados Eliminar

100%

- ✓ Sólo debe de aparecer los mojes que aún se encuentran pendientes por fabricar como producto terminado.
- ✓ Por cada mojes listado, debe de aparecer al siguiente información: Fórmula, Fecha de creación y el Peso en libras.
- ✓ Filtro para buscar un moje sin terminar por nombre de la fórmula.
- ✓ Filtro para buscar un moje sin terminar por fecha de creación.
- ✓ Botón regresar.
- ✓ Al dar clic en el botón regresar este debe de direccionar al módulo de producción.

Añada un elemento

**AÑADIR A LA TARJETA**

- Miembros
- Etiquetas
- Checklist
- Vencimiento
- Adjunto
- Portada

**POWER-UPS**

Conseguir Power-Ups

Consiga Power-Ups ilimitados y mucho más.

Actualizar el equipo

**ACCIONES**

- Mover

Figura 7.1.96

Autor: Auditoría propia, herramienta usada: Trello.

- HU 97

The screenshot shows a Trello card titled "HU - 97. Implementar opciones al seleccionar un moje sin terminar." located in the "Sprint 9" list. The card is divided into several sections:

- Descripción:** Contains a main description, a list of options to appear, and a "Nota" section with two bullet points.
- Criterios de aceptación:** A section with a 100% progress bar and two checked items.
- Right sidebar:** Includes "AÑADIR A LA TARJETA" (Members, Labels, Checklist, Due Date, Attachment, Cover), "POWER-UPS" (Get Power-Ups, Upgrade team), and "ACCIONES" (Move, Copy).

**Descripción** Editar

Yo como administrador del sistema requiero que la aplicación permita seleccionar un moje sin terminar para fabricarlo como producto final.

Al seleccionar el moje sin terminar debe de aparecer las siguientes opciones:

- Registrar producto terminado.
- Ver información.

Nota:

- La funcionalidad del botón para registrar producto terminado fue desarrollado en la HU - 91.
- La funcionalidad para ver la información detallada del moje sin terminar fue desarrollada en la HU - 92.

**Criterios de aceptación** Ocultar elementos completados Eliminar

100% 100%

- Al dar clic en un moje sin terminar se debe de mostrar las siguientes opciones: Registrar producto terminado y Ver información.
- Al dar clic en el botón "Registrar producto terminado", este debe de direccionar a la vista de Registrar producto terminado.

Añada un elemento

**AÑADIR A LA TARJETA**

- Miembros
- Etiquetas
- Checklist
- Vencimiento
- Adjunto
- Portada

**POWER-UPS**

Conseguir Power-Ups

Consiga Power-Ups ilimitados y mucho más.

- Actualizar el equipo

**ACCIONES**

- Mover
- Copiar

Figura 7.1.97

Autor: Auditoría propia, herramienta usada: Trello.

- HU 98

**HU - 98. Vista de las sedes de una empresa.**  
en la lista [Sprint 9](#)

**Descripción** Editar

Yo como administrador del sistema requiero que la aplicación permita ver las sedes que tiene una empresa para seleccionar una de ella e ingresar a sus módulos de:

- Producción
- Talento humano
- Compras e inventario.

Cada Sede debe de manejar su propia información, es decir, la aplicación debe de separar todo por sede. Por ejemplo: Las personas de la sede X no deben de aparecer en las personas de la sede Y. Así se debe de comportar cada módulo.

**Criterios de aceptación** Ocultar elementos completados Eliminar

100%

- ✓ El sistema debe de mostrar la siguiente información por cada sede: Nombre de la sede y las opciones para: Botón para Ver la información detallada, botón de opciones y botón de ingresar.;
- ✓ Al dar clic en el botón "Información", este debe de direccionar a la vista de ver información detallada de la sede.
- ✓ Al dar clic en el botón "Opciones", este debe de abrir una nueva ventana con la opción de editar. Al dar clic en el "Editar" este debe de direccionar a la vista de editar empresa sede.
- ✓ Al dar clic en el botón "Ingresar", este debe de direccionar a la vista de ver información detallada del menú principal.
- ✓ La ingresar a una sede, se debe de garantizar que la información es propia de esa sede, no debe de mezclar datos de otras sede o otras empresas.

Añada un elemento

**AÑADIR A LA TARJETA**

- Miembros
- Etiquetas
- Checklist
- Vencimiento
- Adjunto
- Portada

**POWER-UPS**

Conseguir Power-Ups

Consiga Power-Ups ilimitados y mucho más.

Actualizar el equipo

**ACCIONES**

- Mover
- Copiar
- Convertir en plant...
- Seguir
- Archivar
- Compartir

Figura 7.1.98

Autor: Auditoría propia, herramienta usada: Trello.

- HU 99

The image shows a Trello card titled "HU - 99. Ver información detallada de una sede." with a sub-header "en la lista Sprint 9". The card is organized into sections: "Descripción" with an "Editar" button, "Criterios de aceptación" with a 100% progress bar and three checked items, and a right-hand sidebar with "AÑADIR A LA TARJETA" options and "POWER-UPS" information.

**Descripción** Editar

Yo como administrador del sistema requiero que la aplicación permita seleccionar una sede para ver su información completa y detallada.

**Criterios de aceptación** Ocultar elementos completados Eliminar

100%

- El sistema debe de mostrar la siguiente información: Nombre de la sede, Contacto, Teléfono, Email, Dirección, Ciudad, Estado.
- Botón regresar.
- Al dar clic en el botón regresar este debe de direccionar a la vista de sedes.

Añada un elemento

**AÑADIR A LA TARJETA**

- Miembros
- Etiquetas
- Checklist
- Vencimiento
- Adjunto
- Portada

**POWER-UPS**

Conseguir Power-Ups

Consiga Power-Ups ilimitados y

Figura 7.1.99

Autor: Auditoría propia, herramienta usada: Trello.

- HU 100

HU - 100. Editar o actualizar la información de una sede.
✕

en la lista [Sprint 9](#)

Descripción Editar

Yo como administrador del sistema requiero que la aplicación permita Editar la información de una sede, para corregir información mal ingresada o para mantener la información actualizada.

Si se cambia el campo de estado a "inactivo", debe de desaparecer el botón de ingresar a una sede.

Criterios de aceptación Ocultar elementos completados Eliminar

100%

- Se debe de mostrar la siguiente información para editar: Nombre, Teléfono, Contacto, Ciudad, Email, Estado, Dirección.
- Si se actualiza el campo "Nombre", sólo permitirá ingresar un nombre que tenga cómo mínimo 4 caracteres.
- Si se actualiza el campo "Teléfono", sólo permitirá ingresar un teléfono que tenga cómo mínimo 6 caracteres.
- Si se actualiza el campo "Contacto", sólo permitirá ingresar un Contacto que tenga cómo mínimo 6 caracteres.
- Si se actualiza el campo "Ciudad", sólo permitirá ingresar una Ciudad que tenga cómo mínimo 4 caracteres.
- Si se actualiza el campo "Email", sólo permitirá ingresar un Email que tenga cómo mínimo 4 caracteres.
- Si se actualiza el campo "Dirección", sólo permitirá ingresar una Dirección que tenga cómo mínimo 4 caracteres.
- Si se actualiza el campo de "Estado", debe de permitir elegir la opción de Activo e Inactivo.
- Botón para guardar los datos actualizados.
- Botón para cancelar el proceso.
- Si el estado fue actualizado a "Inactivo", al observar la sede, el botón de "Ingresar" debe de desaparecer.
- Al observar la sede, los datos editados deben de quedar actualizados correctamente.

Añada un elemento

**AÑADIR A LA TARJETA**

- Miembros
- Etiquetas
- Checklist
- Vencimiento
- Adjunto
- Portada

**POWER-UPS**

Conseguir Power-Ups

Consiga Power-Ups ilimitados y mucho más.

Actualizar el equipo

**ACCIONES**

- Mover
- 📄 Copiar
- 🏠 Convertir en plant...
- 👁️ Seguir
- 📁 Archivar
- 🔗 Compartir

Figura 7.1.100

Autor: Auditoría propia, herramienta usada: Trello.

- HU 101

The image shows a Trello card titled "HU - 101. Vista de la empresa." located in the "Sprint 9" list. The card content includes a description, a list of requirements, and acceptance criteria. The progress bar is at 100%. On the right, there are sidebar options for adding items to the card and power-ups.

**HU - 101. Vista de la empresa.**  
en la lista [Sprint 9](#)

**Descripción** [Editar](#)

Yo como administrador del sistema requiero que la aplicación permita ver la Empresa que tengo, para:

- Ingresar a sus sedes
- Actualizar su información.
- Ver su información completa

Sólo debe de aparecer la empresa que yo como administrador tengo asignada.

**Criterios de aceptación** [Ocultar elementos completados](#) [Eliminar](#)

100%

- El sistema debe de mostrar el nombre de la empresa y una imagen que la represente.
- Sólo debe de aparecer la empresa que el administrador tenga asignada.

[Añada un elemento](#)

**AÑADIR A LA TARJETA**

- [Miembros](#)
- [Etiquetas](#)
- [Checklist](#)
- [Vencimiento](#)
- [Adjunto](#)
- [Portada](#)

**POWER-UPS**

[Conseguir Power-Ups](#)

Consiga Power-Ups ilimitados y mucho más.

[Actualizar el equipo](#)

**ACCIONES**

Figura 7.1.101

Autor: Auditoría propia, herramienta usada: Trello.

- HU 102

The image shows a Trello card titled "HU - 102. Implementar opciones al seleccionar una empresa." located in the "Sprint 9" list. The card is organized into several sections:

- Descripción:** A text block explaining the requirement: "Yo como administrador del sistema requiero que la aplicación permita seleccionar y muestre las siguientes opciones para seleccionar:" followed by a bulleted list:
  - Ver la información de la empresa seleccionada.
  - Editar o actualizar la información de la empresa seleccionada.
  - Ingresar a las sedes de la empresa.
- Criterios de aceptación:** A section with a 100% progress bar and four checklist items, each with a blue checkmark:
  - ✓ Al seleccionar la empresa el sistema debe de mostrar dos opciones: Ver información, Editar e Ingresar.
  - ✓ Al dar clic en el botón "Ver información" este debe de direccionar a la vista de ver información detallada de la empresa.
  - ✓ Al dar clic en el botón "Editar" este debe de direccionar a la vista de Actualizar empresa
  - ✓ Al dar clic en el botón "Ingresar" este debe de direccionar a la vista de las sedes de la empresa.
- Acciones:** A button labeled "Añada un elemento" at the bottom left.
- Panel lateral derecho:** Contains utility options:
  - AÑADIR A LA TARJETA:** Buttons for "Miembros", "Etiquetas", "Checklist", "Vencimiento", "Adjunto", and "Portada".
  - POWER-UPS:** A "Conseguir Power-Ups" button and a note "Consiga Power-Ups ilimitados y mucho más." with a button "Actualizar el equipo".
  - ACCIONES:** Buttons for "Mover" and "Copiar".

Figura 7.1.102

Autor: Auditoría propia, herramienta usada: Trello.

- HU 103

The image shows a Trello card titled "HU - 103. Ver información detallada de una empresa seleccionada." located in the "Sprint 9" list. The card is organized into several sections:

- Descripción:** A text block stating, "Yo como administrador del sistema requiero que la aplicación permita ver la información detallada de la empresa seleccionada, para consultar su información en caso de que se requiera." There is an "Editar" button next to the title.
- Criterios de aceptación:** A section with a 100% progress bar and three checked items:
  - ✓ El sistema debe de mostrar la siguiente información: Nombre de la empresa, Nit, Contacto, Teléfono, Email, Página Web, Estado.
  - ✓ Botón regresar.
  - ✓ Al dar clic en el botón regresar este debe de direccionar a la vista de la empresa.Buttons for "Ocultar elementos completados" and "Eliminar" are present.
- Añadir a la tarjeta:** A sidebar on the right with buttons for "Miembros", "Etiquetas", "Checklist", "Vencimiento", "Adjunto", and "Portada".
- Power-Ups:** A section at the bottom right with a "Conseguir Power-Ups" button and a note: "Consiga Power-Ups ilimitados y mucho más."

Figura 7.1.103

Autor: Auditoría propia, herramienta usada: Trello.

- HU 104

**HU - 104. Editar actualizar la información de una empresa seleccionada.**  
 en la lista [Sprint 9](#)

**Descripción** Editar

Yo como administrador del sistema requiero que la aplicación permita Editar la información de una empresa, para corregir información mal ingresada o para mantener la información actualizada.

Si se cambia el campo de estado a "inactivo", debe de desaparecer el botón de ingresar a las sedes.

**Criterios de aceptación** Ocultar elementos completados Eliminar

100%

- ✓ Se debe de mostrar la siguiente información para editar: Nombre, NIT, Teléfono, Página web, Contacto, Email, Estado;
- ✓ Si se actualiza el campo "Nombre", sólo permitirá ingresar un nombre que tenga cómo mínimo 4 caracteres;
- ✓ Si se actualiza el campo "NIT", sólo permitirá ingresar un NIT que tenga cómo mínimo 5 caracteres;
- ✓ Si se actualiza el campo "Teléfono", sólo permitirá ingresar un teléfono que tenga cómo mínimo 6 caracteres;
- ✓ Si se actualiza el campo "Página web", sólo permitirá ingresar una dirección de pagina que tenga cómo mínimo 4 caracteres;
- ✓ Si se actualiza el campo "Contacto", sólo permitirá ingresar un Contacto que tenga cómo mínimo 6 caracteres;
- ✓ Si se actualiza el campo "Email", sólo permitirá ingresar un Email que tenga cómo mínimo 4 caracteres;
- ✓ Si se actualiza el campo de "Estado", debe de permitir elegir la opción de Activo e Inactivo;
- ✓ Si el estado fue actualizado a "Inactivo", al observar la empresa, el botón de "Ingresar" debe de desaparecer;
- ✓ Botón para guardar los datos actualizados;
- ✓ Botón para cancelar el proceso;
- ✓ Al observar la empresa, los datos editados deben de quedar actualizados correctamente;

Añada un elemento

**AÑADIR A LA TARJETA**

- Miembros
- Etiquetas
- Checklist
- Vencimiento
- Adjunto
- Portada

**POWER-UPS**

Conseguir Power-Ups

Consiga Power-Ups ilimitados y mucho más.

Actualizar el equipo

**ACCIONES**

- Mover
- Copiar
- Convertir en plant...
- Seguir
- Archivar
- Compartir

Figura 7.1.104

Autor: Auditoría propia, herramienta usada: Trello.

- HU 105

**HU - 105. Implementar menú(Navbar) y footer.**  
en la lista [Sprint 9](#)

**Descripción** Editar

Yo como administrador del sistema requiero que la aplicación tengo una barra de menú con las siguientes opciones:

- Logotipo de la aplicación
- Sedes
- Producción
- Compras e inventario
- Personas
- perfil de usuario.

Para ingresar ir de manera directa a los módulos principales de la aplicación.

**Criterios de aceptación** Ocultar elementos completados Eliminar

100%

- ✓ Al dar clic en "sedes", este debe de direccionar a las sedes de la empresa.
- ✓ Al dar clic en "Producción" este debe direccionar al módulo de producción.
- ✓ Al dar clic en "Compras e inventario" este debe direccionar al módulo de Compras e inventario.
- ✓ Al dar clic en "Personas" este debe direccionar al módulo de Talento humano.
- ✓ El botón de "perfil de usuario" debe de aparecer una imagen o silueta de usuario.
- ✓ Al dar clic en la imagen de usuario este debe de desprender dos opciones: 1. Ver perfil. 2. Salir de la aplicación.
- ✓ Al dar clic en "Ver perfil", este debe de direccionar a la vida de ver perfil de usuario.
- ✓ Al dar clic en "Salir", el usuario debe de salir de la aplicación y direccionarlo a la vista a de Iniciar sesión.
- ✓ Imagen con el logotipo de la aplicación.

Añada un elemento

**AÑADIR A LA TARJETA**

- Miembros
- Etiquetas
- Checklist
- Vencimiento
- Adjunto
- Portada

**POWER-UPS**

Conseguir Power-Ups

Consiga Power-Ups ilimitados y mucho más.

- Actualizar el equipo

**ACCIONES**

- Mover
- Copiar
- Convertir en plant...
- Seguir
- Archivar
- Compartir

Figura 7.1.105

Autor: Auditoría propia, herramienta usada: Trello.

- HU 106

The image shows a Trello card titled "HU - 106. Ver perfil de usuario." located in the "Sprint 9" list. The card is currently in the "Descripción" tab, with an "Editar" button next to it. The description text reads: "Yo como administrador del sistema requiero que la aplicación permita ver la información detallada del mi perfil de usuario del empleado que se esté utilizando la aplicación, para consultarla cada vez que se necesite." Below the description is a "Criterios de aceptación" section with a 100% progress bar and three checked items: "El sistema debe de mostrar la siguiente información: Nombre del empleado, Empresa para que trabaja, Usuario de la aplicación, Rol o cargo y la fecha de ingreso.", "botón 'Actualizar contraseña'", and "Al dar clic en el botón de 'Actualizar contraseña' este debe de direccionar a la vista de actualizar credenciales." There are buttons for "Ocultar elementos completados" and "Eliminar" next to the criteria section, and an "Añada un elemento" button at the bottom. On the right side of the card, there is a sidebar with "AÑADIR A LA TARJETA" options: "Miembros", "Etiquetas", "Checklist", "Vencimiento", "Adjunto", and "Portada". Below that is a "POWER-UPS" section with a "Conseguir Power-Ups" button and a note: "Consiga Power-Ups ilimitados y mucho más." with a link to "Actualizar el equipo".

Figura 7.1.106

Autor: Auditoría propia, herramienta usada: Trello.

- HU 107

**HU - 107. Asignar credenciales de usuario a un empleado.**  
en la lista [Sprint 9](#)

**Descripción** Editar

Yo como administrador del sistema requiero que la aplicación permita a un Empleado con Rol de administrador o Super administrador asignar las credenciales a un empleado.

En la vista de ver información detallada del empleado, requiero de un botón que permita asignar las credenciales a ese usuario seleccionado.

Las credenciales son:

- Usuario
- Contraseña

Estas credenciales se deben de construir de la siguiente manera:

- Usuario: Nombre del empleado + . + primer apellido + último 4 dígitos del número de identificación.
- Contraseña: Es el número de identificación del empleado.

Estas credenciales serán las que permitirán al empleado ingresar a la aplicación.

Si se consulta la información de un empleado y este ya tiene usuario asignado, entonces, no debe de aparecer el botón para asignar usuario.

**Criterios de aceptación** Ocultar elementos completados Eliminar

100%

- Botón para asignar credenciales al empleado seleccionado.
- Si el empleado seleccionado ya tiene usuario asignado, entonces no debe de aparecer el botón para asignar credenciales.
- Al dar clic en el botón asignar credenciales, debe de aparecer un mensaje de confirmación. Si se da Aceptar, el sistema debe de crear las credenciales al empleado.
- Se debe de crear el usuario con base a la regla de la descripción de la HU.
- Se debe de crear la contraseña con base a la regla definida en la descripción de la HU.

Añada un elemento

**AÑADIR A LA TARJETA**

- Miembros
- Etiquetas
- Checklist
- Vencimiento
- Adjunto
- Portada

**POWER-UPS**

Conseguir Power-Ups

Consiga Power-Ups ilimitados y mucho más.

[Actualizar el equipo](#)

**ACCIONES**

- Mover
- Copiar
- Convertir en plant...
- Seguir
- Archivar
- Compartir

Figura 7.1.107

Autor: Auditoría propia, herramienta usada: Trello.

## Sprint 10

- HU 108

**HU - 108. Implementar login para el acceso a la aplicación.**  
en la lista [Sprint 10](#)

**Descripción** Editar

Yo como administrador del sistema requiero que la aplicación tenga la vista de Login para poder acceder a la aplicación. El login debe de solicitar la siguiente información:

- Usuario.
- Contraseña.

Si las credenciales son correctas, el empleado podrá ingresar a la aplicación, de lo contrario el sistema mostrará un mensaje informando que las credenciales son erradas.

Además, requiero que halla un botón para recuperar las credenciales en caso de que el empleado olvide su usuario o contraseña. Este botón desplegará un mensaje que le informará al empleado lo siguiente:

"¿Olvidaste tus credenciales?  
Para recuperar tu contraseña debes de contactar al administrador de tu sede, él recuperará tus credenciales."

**Criterios de aceptación** Ocultar elementos completados Eliminar

100%

- ✓ Campo para digitar el usuario.
- ✓ Campo para digitar la contraseña.
- ✓ Botón para ingresar a la aplicación.
- ✓ Botón para recuperar las credenciales.
- ✓ Al dar clic en "Recuperar", el sistema mostrará un mensaje con la información descrita en la HU.
- ✓ En caso de las credenciales sean incorrectas, se debe de mostrar el siguiente mensaje: Usuario o contraseña incorrecto
- ✓ Ambos campos deben de ser obligatorios.

Añada un elemento

**AÑADIR A LA TARJETA**

- Miembros
- Etiquetas
- Checklist
- Vencimiento
- Adjunto
- Portada

**POWER-UPS**

Conseguir Power-Ups

Consiga Power-Ups ilimitados y mucho más.

Actualizar el equipo

**ACCIONES**

- Mover
- Copiar
- Convertir en plant...
- Seguir
- Archivar
- Compartir

Figura 7.1.108

Autor: Auditoría propia, herramienta usada: Trello.

- HU 109

**HU- 109. Actualizar credenciales de usuario.**  
en la lista [Sprint 10](#)

**Descripción** Editar

Yo como administrador del sistema requiero que la aplicación permita a un empleado actualizar su contraseña de ingreso a la aplicación, para que el empleado pueda cambiar su contraseña constantemente y evitar robos de identidad.

El botón para actualizar la contraseña se realizó en la HU- 105m que diseccionará a esta vista que permitirá actualizar la contraseña.

**Criterios de aceptación** Ocultar elementos completados Eliminar

100%

- El sistema debe de mostrar la siguiente información: usuario de la aplicación y un campo vacío solicitando ingresar la nueva contraseña.
- El campo "Contraseña nueva" no puede estar vacío.
- Botón para guardar la nueva contraseña.
- Botón para cancelar el proceso.
- Si se da en el botón guardar, el usuario tiene que quedar con la nueva contraseña asignada.
- Sólo se podrá cambiar al contraseña, el usuario si se debe de conservar siempre.

Añada un elemento

**AÑADIR A LA TARJETA**

- Miembros
- Etiquetas
- Checklist
- Vencimiento
- Adjunto
- Portada

**POWER-UPS**

Conseguir Power-Ups

Consiga Power-Ups ilimitados y mucho más.

Actualizar el equipo

**ACCIONES**

- Mover
- Copiar
- Convertir en plant...

Figura 7.1.109

Autor: Auditoría propia, herramienta usada: Trello.

- HU 110

**HU - 110. Recuperar credenciales de usuario.**  
en la lista [Sprint 10](#)

**Descripción** Editar

Yo como administrador del sistema requiero que la aplicación permita a un empleado con rol de administrador o rol de super administrador recuperar las credenciales a un empleado.

En la vista de "ver información detallada de un empleado", requiero de un botón que permita ir a recuperar las credenciales del empleado. Este botón sólo estará disponible para empleados que tengan ya un usuario asignado, de lo contrario no se debe de mostrar el botón.

Este botón deberá direccionar a la vista de recuperar credenciales.

**Criterios de aceptación** Ocultar elementos completados Eliminar

100%

- ✓ Botón "Recuperar credenciales", En la vista de "ver información detallada de un empleado". Si el empleado no tiene usuario asignado, el botón "Recuperar credenciales" no debe de aparecer.
- ✓ En la vista de Recuperar contraseña: El sistema debe de mostrar el nombre del usuario del empleado y solicitar en el campo de "ingresar nueva contraseña" la contraseña.
- ✓ Botón para guardar la nueva contraseña.
- ✓ Botón para cancelar el proceso.
- ✓ El empleado debe de poder ingresar a la aplicación con la nueva contraseña que le asignó el administrador.

Añada un elemento

**AÑADIR A LA TARJETA**

- Miembros
- Etiquetas
- Checklist
- Vencimiento
- Adjunto
- Portada

**POWER-UPS**

Conseguir Power-Ups

Consiga Power-Ups ilimitados y mucho más.

Actualizar el equipo

**ACCIONES**

- Mover
- Copiar
- Convertir en plant...
- Seguir

Figura 7.1.110

Autor: Auditoría propia, herramienta usada: Trello.

- HU 111

**HU - 111. Permisos de usuario de acuerdo al Rol del empleado.**  
en la lista [Sprint 10](#)

**Descripción** Editar

Yo como administrador del sistema requiero que la aplicación permita el accesos a los diferentes módulos de acuerdo al rol del empleado, para segregar las funciones de acuerdo al rol.

Los roles que tendrá el sistema son:

- Super administrador
- Administrador
- Panadero

Los permisos del super administrador serán:

- Este tipo de rol tendrá permiso a todas las funcionalidades de la aplicación.

Los permisos del administrador serán:

- No tendrá acceso a empresas y sedes, pero tendrá todos lo permisos para los módulos de producción, talento humano y compras e inventario de la sede a la cual pertenezca.

Los permisos del rol de panadero serán:

- Módulo de producción: Puede usar todas las funciones del módulo de producción para consultar. Tampoco tendrá permisos para actualizar información. Este rol solo podrá crear en las opciones de mojes y de productos terminados.
- Módulo de compras e inventario: Sólo podrá consultar, no tiene permisos para ingresar información, editar o eliminar.
- No tendrá acceso a: Empresas, Sedes, ni a el módulo de talento humano.

**Criterios de aceptación** Ocultar elementos completados Eliminar

100%

- ✓ Cuando ingrese un empleado con el rol de Panadero, el sistema debe de llevarlo directamente a la vista de producción. En el Navbar no pedrán aparecer las opciones de Sedes y Talento humano. Por parte del módulo de comparas e inventario este solo debe de tener permisos para consultar. Por parte del módulo de producción este rol solo pedrá consultar, y crear el la opción de mojes y producto terminados.
- ✓ Cuando ingrese un empleado con el rol de Administrador el sistema debe de llevarlo directamente al menú principal. Debe de tener acceso a todas las opciones del menú principal. No debe de aparecer en el Navbar la opción de sedes.
- ✓ Cuando ingrese un empleado con Rol de Super-Administrador, el sistema debe de llevarlo directamente a la empresa a la cual pertenece. Debe de tener acceso a todos los módulos opciones de la aplicación.

Añada un elemento

**AÑADIR A LA TARJETA**

- Miembros
- Etiquetas
- Checklist
- Vencimiento
- Adjunto
- Portada

**POWER-UPS**

Conseguir Power-Ups

Consiga Power-Ups ilimitados y mucho más.

Actualizar el equipo

**ACCIONES**

- Mover
- Copiar
- Convertir en plant...
- Seguir
- Archivar
- Compartir

Figura 7.1.111

Autor: Auditoría propia, herramienta usada: Trello.

- HU 112

The image shows a Trello card titled "HU - 112. Deshabilitar un empleado." located in the "Sprint 10" list. The card contains a description, acceptance criteria, and a progress bar. On the right side, there are sidebar options for adding items to the card and power-ups.

**HU - 112. Deshabilitar un empleado.**  
en la lista [Sprint 10](#)

**Descripción** Editar

Yo como administrador del sistema requiero que la aplicación permita dar de baja a las credenciales de un empleado, para quitar el ingreso a la aplicación en caso de que el empleado no trabaje mas para la empresa.

Cuando se seleccione un empleado para actualizar su información y se cambie el estado del empleado de Activo a Inactivo, el sistema debe de borrar las credenciales de usuario de ese empleado seleccionado.

**Criterios de aceptación** Ocultar elementos completados Eliminar

100%

- Si se desactiva un empleado, es decir, su estado es false, el sistema debe de borrar sus credenciales.
- Si se intenta ingresar a la aplicación con las credenciales de ese empleado, el sistema no debe permitirlo.

Añada un elemento

**AÑADIR A LA TARJETA**

- Miembros
- Etiquetas
- Checklist
- Vencimiento
- Adjunto
- Portada

**POWER-UPS**

Conseguir Power-Ups

Consiga Power-Ups ilimitados y mucho más.

Actualizar el equipo

Figura 7.1.112

Autor: Auditoría propia, herramienta usada: Trello.

- HU 113

**HU - 113. Notificación de correos al empleado.**  
en la lista [Sprint 10](#)

**Descripción** Editar

Yo como administrador del sistema requiero que la aplicación notifique con un correo al empleado cuando:

- Se creen las credenciales al empleado.
- El empleado actualice su contraseña.
- El administrador recupere las credenciales de un empleado.

Para informar al empleado cada vez que se haga alguno de estos movimientos.

**Criterios de aceptación** Ocultar elementos completados Eliminar

100%

- Cuando se asignen las credenciales a un empleado debe de llegar un mensaje al correo del empleado con la siguiente información: Bienvenido!! Su usuario fue creado con éxito!! Hola "NOMBRE DEL EMPLEADO" la Empresa "NOMBRE DE LA EMPRESA" Sede "NOMBRE DE LA SEDE" te da la bienvenida; a continuación tus credenciales de usuario: USUARIO: usuario del empleado generado por el sistema. CONTRASEÑA: Debes ingresar tú número de documento de identidad
- Cuando un empleado actualice su contraseña, debe de llegar una notificación al correo del empleado con la siguiente información: Actualizar Contraseña!! Hola "USUARIO DEL EMPLEADO" Su contraseña ha sido actualizada con éxito Si cambiaste tu contraseña, hacer caso omiso a este correo, de lo contrario favor contactar al administrador de tu empresa.
- Cuando un administrador recupere las credenciales a un empleado, debe de llegar una notificación al correo del empleado con la siguiente información: Recuperar Credenciales!! Hola "USUARIO DEL EMPLEADO" Su administrador ha recuperado su contraseña con éxito Favor contactar al administrador de tu empresa. ¡Saludos!

Añade un elemento

**AÑADIR A LA TARJETA**

- Miembros
- Etiquetas
- Checklist
- Vencimiento
- Adjunto
- Portada

**POWER-UPS**

Conseguir Power-Ups

Consiga Power-Ups ilimitados y mucho más.

[Actualizar el equipo](#)

**ACCIONES**

- Mover
- Copiar
- Convertir en plant...
- Seguir
- Archivar
- Compartir

Figura 7.1.113

Autor: Auditoría propia, herramienta usada: Trello.

- HU 114

**HU - 114. Notificación de correo de inventario bajo.**  
en la lista [Sprint 10](#)

**Descripción** Editar

Yo como administrador del sistema requiero que la aplicación notifique al correo de los Super administradores y administradores de una sede, cuando:

- Materias primas del inventario se estén agotando. Es decir, las materias primas que tengan una cantidad menos a la cantidad mínima de inventario. La notificación debe de generarse empezando el día laboral 5 am y terminando el día laboral 9 pm. (Hora Bogotá Colombia)
- Además, requiero que en la vista de inventario bajo de materia prima, se agregue un botón para notificar a los administradores el inventario bajo.

Al correo debe llegar la información de las materias primas que están bajas, para tener informado a los administradores que deben de comprar para garantizar la producción de la panadería.

**Criterios de aceptación** Ocultar elementos completados Eliminar

100%

- ✓ La notificación debe de llegar a todos los Super administradores y administradores de la sede.
- ✓ Deben de llegar por día dos correos, el de las 5 am y el de las 9 pm de forma automática.
- ✓ Botón en la vista de inventario bajo para notificar por medio de un correo a los administradores el inventario bajo.
- ✓ El cuerpo del correo debe de tener la siguiente información por cada materia prima reportada como baja: Nombre de la materia prima, Cantidad actual en el inventario, Cantidad de inventario mínimo, Email del proveedor de la materia prima y teléfonos de contacto del proveedor de la materia prima.

Añada un elemento

**AÑADIR A LA TARJETA**

- Miembros
- Etiquetas
- Checklist
- Vencimiento
- Adjunto
- Portada

**POWER-UPS**

Conseguir Power-Ups

Consiga Power-Ups ilimitados y mucho más.

Actualizar el equipo

**ACCIONES**

- Mover
- Copiar
- Convertir en plant...
- Seguir
- Archivar
- Compartir

Figura 7.1.114

Autor: Auditoría propia, herramienta usada: Trello.

- HU 115

**HU - 115. Notificación de correo por falta de insumos para crear un moje.**  
en la lista [Sprint 10](#)

**Descripción** Editar

Yo como administrador del sistema requiero que la aplicación permita a un empleado notificar vía correo a los super administradores y administradores de la sede, cuando no se pueda fabricar un moje por falta de insumos.

Requiero de un botón en la vista de Fabricar moje para notificar a los administradores.

Nota: El empleado podrá hacer uso de este botón cuando lo considere pertinentemente.

**Criterios de aceptación** Ocultar elementos completados Eliminar

100%

- ✓ Botón para notificar al administrador que no se pudo fabricar el moje por falta de insumos.
- ✓ Al dar clic en el botón de notificar debe de enviar el correo a los super administradores y administradores.
- ✓ El cuerpo del correo debe de tener la siguiente información por cada insumo reportado como bajo: Nombre de la materia prima, Cantidad actual en el inventario, Cantidad de inventario mínimo, Email del proveedor de la materia prima y teléfonos de contacto del proveedor de la materia prima;

Añada un elemento

**AÑADIR A LA TARJETA**

- Miembros
- Etiquetas
- Checklist
- Vencimiento
- Adjunto
- Portada

**POWER-UPS**

Conseguir Power-Ups

Consiga Power-Ups ilimitados y mucho más.

Actualizar el equipo

**ACCIONES**

- Mover
- Copiar
- Convertir en plant...

Figura 7.1.115

Autor: Auditoría propia, herramienta usada: Trello.

- HU 116

The image shows a Trello card titled "HU - 116. Adaptar la aplicación a cualquier tamaño de pantalla." located in the "Sprint 10" list. The card contains a description, acceptance criteria, and a progress bar. On the right side, there are several utility sections: "AÑADIR A LA TARJETA" with options like "Miembros", "Etiquetas", "Checklist", "Vencimiento", "Adjunto", and "Portada"; "POWER-UPS" with a "Conseguir Power-Ups" button and a link to "Actualizar el equipo"; and "ACCIONES" with a "Mover" button.

**HU - 116. Adaptar la aplicación a cualquier tamaño de pantalla.**  
en la lista [Sprint 10](#)

**Descripción** Editar

Yo como administrador del sistema requiero que la aplicación se adapte a cualquier tamaño de pantalla sin importar si es un ordenador, table o celular, la aplicación debe de poder funcionar correctamente, para tener una buena experiencia en el uso de la aplicación.

**Criterios de aceptación** Ocultar elementos completados Eliminar

100%

- Si la aplicación se abre en un celular debe de funcionar correctamente.
- Si la aplicación se abre en un ordenador debe de funcionar correctamente.
- Si la aplicación se abre en una tableta debe de funcionar correctamente.
- Todas las vistas del aplicativo se deben de adaptar al tamaño de la pantalla del dispositivo donde se esta usando la aplicación.

Añada un elemento

**AÑADIR A LA TARJETA**

- Miembros
- Etiquetas
- Checklist
- Vencimiento
- Adjunto
- Portada

**POWER-UPS**

Conseguir Power-Ups

Consiga Power-Ups ilimitados y mucho más.

Actualizar el equipo

**ACCIONES**

- Mover

Figura 7.1.116

Autor: Auditoría propia, herramienta usada: Trello.

- HU 117



The image shows a Trello card titled "HU - 117. Pruebas de servicios de backend con Postman." located in the "Sprint 10" list. The card has a description, acceptance criteria, and a progress bar showing 100% completion. On the right side, there are options to add the card to a board, such as "Miembros", "Etiquetas", "Checklist", "Vencimiento", "Adjunto", and "Portada".

**HU - 117. Pruebas de servicios de backend con Postman.**  
en la lista [Sprint 10](#)

**Descripción** Editar

Se deben de probar cada servicio construido para el correcto funcionamiento de la aplicación. Se necesita testear cada servicio con la aplicación de Postman con el fin de que cada servicio arroje la información que se espera.

**Criterios de aceptación** Ocultar elementos completados Eliminar

100%

- Cada servicio de la aplicación debe de ser probado.
- El resultado que arroje postman debe de coincidir con lo esperado.

Añada un elemento

**AÑADIR A LA TARJETA**

- Miembros
- Etiquetas
- Checklist
- Vencimiento
- Adjunto
- Portada

**POWER-UPS**

Figura 7.1.117

Autor: Auditoría propia, herramienta usada: Trello.

## 7.2 Diseño del Backend

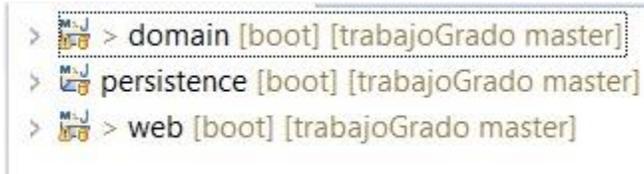


Figura 7.2.1

Autor: Autoría propia. Backend.

El backend se encuentra desarrollado en tres proyectos, donde cada proyecto corresponde a las tres capas que se definieron en la arquitectura, la capa de dominio, la capa de persistencia y la capa web.

### Capa de Dominio

En la capa de dominio se tiene lo siguiente:

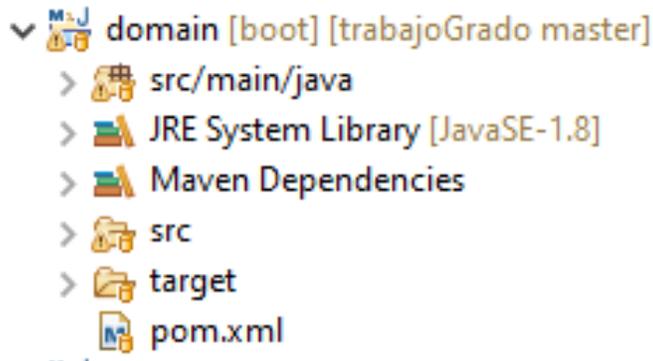


Figura 7.2.2

Autor: Autoría propia.

En la carpeta de 'domain' se encuentran 5 paquetes y un archivo, cada paquete contiene más carpetas internas, donde se encuentra el código de la aplicación y librerías. El archivo 'pom.xml' es el administrador de dependencias, donde se ponen las dependencias del proyecto.

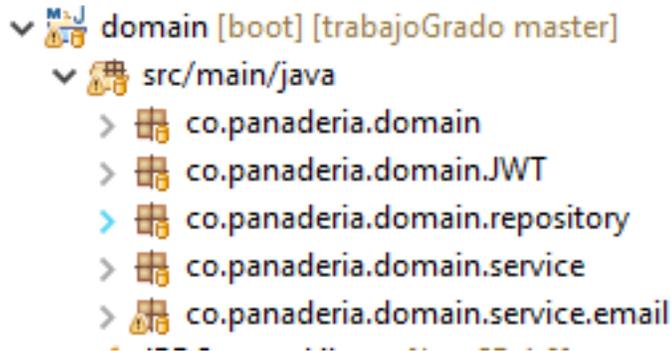
*Paquete src/main/java*

Figura 7.2.3

Autor: Autoría propia.

El Paquete 'src/main/java' contiene el código fuente de la capa de dominio. Este paquete contiene otros paquetes internos, donde cada uno de ellos tienen responsabilidades específicas para el correcto funcionamiento de la capa de dominio. El primer paquete que se observa en la Figura 7.2.4 llamado 'co.panaderia.domain' se encuentran las entidades de dominio.

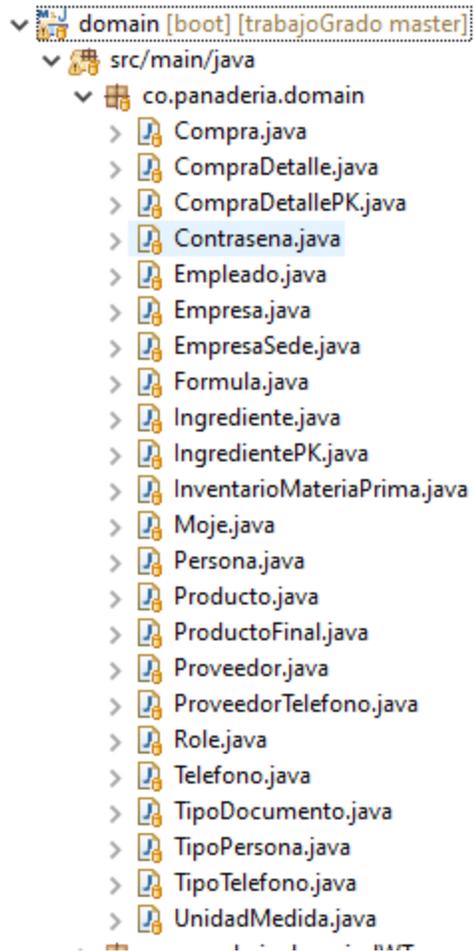


Figura 7.2.4

Autor: Autoría propia.

Cada clase de dominio contiene los atributos, constructores y métodos de getters and setters.

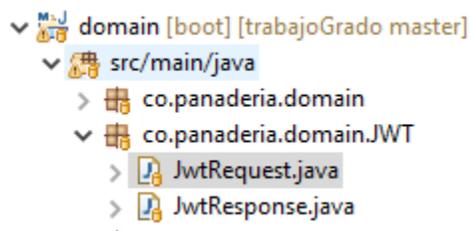


Figura 7.2.5

Autor: Autoría propia.

En el segundo paquete ‘co.panaderia.domain.JWT’ Figura 7.2.5 contiene dos clases que se usan como dominio de entidades para la generación del usuario y contraseña de la seguridad implementada con Jeison Web Token.

La primera clase ‘JwtRequest.java’ contiene los atributos, métodos y getter and setters para el usuario y contraseña de JWT.

La segunda clase ‘JwtResponse.java’ contiene la lógica de declaración de un token con su respectivo get y set. Esto es requerido para el correcto funcionamiento de la seguridad implementada con Jeison web Token.

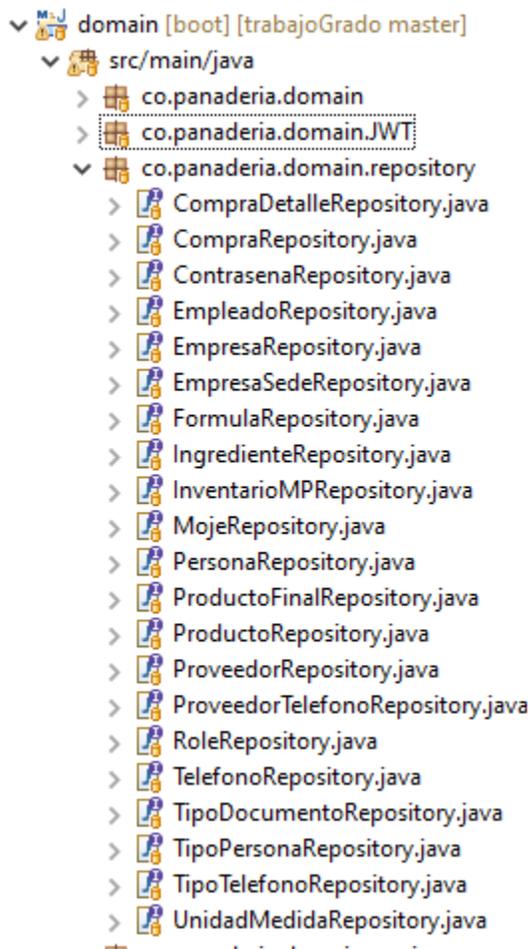


Figura 7.2.6

Autor: Autoría propia.

En el tercer paquete ‘co.panaderia.domain.repository’ Figura 7.2.6 se encuentran las interfaces correspondientes a cada una de la entidades de dominio. Estas interfaces contienen los métodos abstractos.

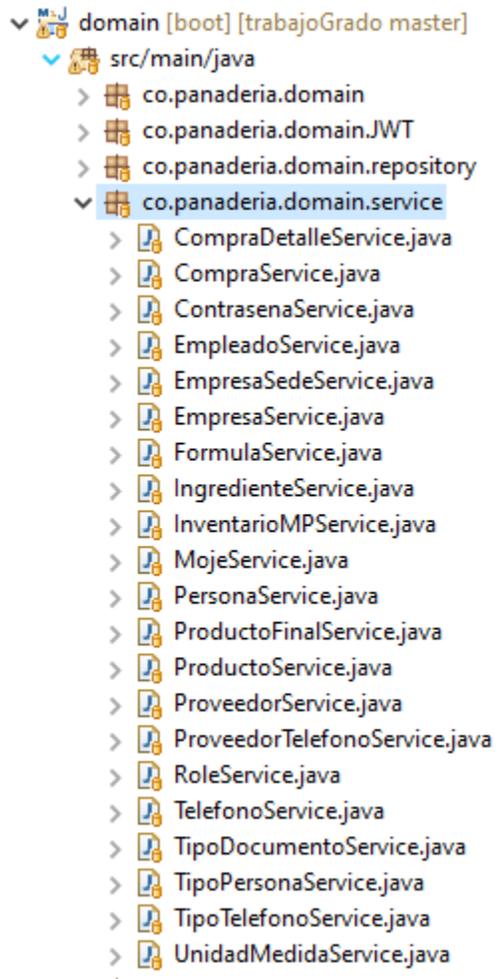


Figura 7.2.7

Autor: Autoría propia.

El cuarto paquete ‘co.panaderia.domain.service’ Figura 7.2.7 contiene los servicios de dominio. Cada clase tipo service hace el papel de intermediario y contienen diferentes métodos, además, se hace uso de inyección de dependencias de las clases de ‘co.panaderia.domain.repository’.

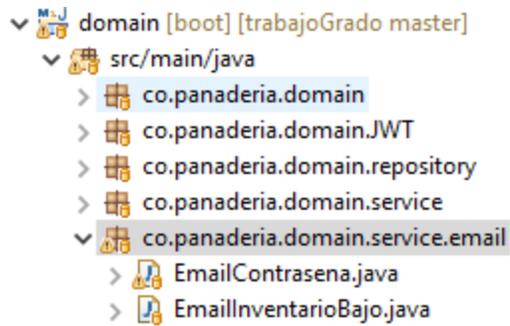


Figura 7.2.8

Autor: Autoría propia.

El quinto paquete ‘co.panaderia.domain.service.email’ Figura 7.2.8 contiene la lógica para el envío de notificaciones de correos a los empleados con todo lo referente a sus credenciales de usuario e inventarios. La primera clase ‘EmailContaseña’ contiene los atributos, métodos, getter y setters de la estructura de un email, como el asunto, el destinatario y el cuerpo del correo. Además, contiene la lógica para el envío de correos de credenciales de usuario. La segunda clase ‘EmailInventarioBajo’ tiene la misma estructura de ‘EmailContaseña’, sin embargo, este se encarga del envío de notificaciones de correos de los inventarios de la panadería.

### Paquete JRE System Library

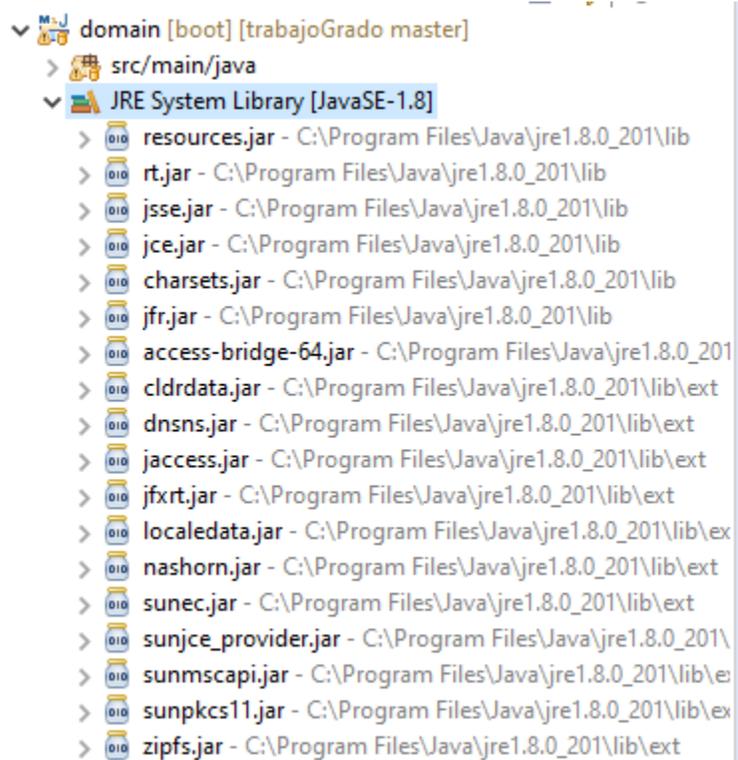


Figura 7.2.9

Autor: Autoría propia. Librerías del framework de Spring.

En este paquete contiene todas las librerías que necesita el sistema para el correcto funcionamiento de la capa de dominio.

## Maven Dependencies

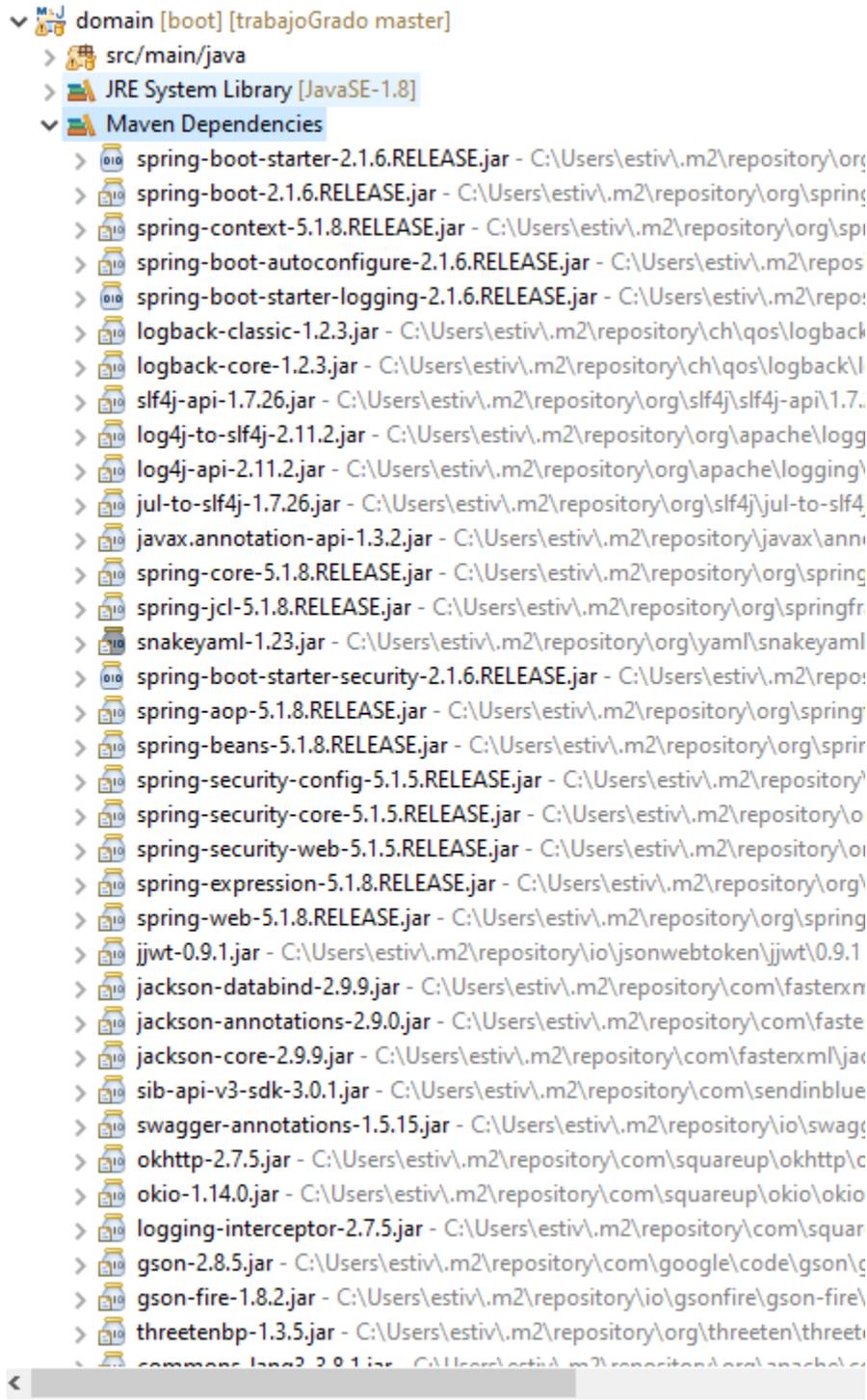


Figura 7.2.10

Autor: Autoría propia. Dependencias del framework de Spring.

En esta carpeta se encuentran las dependencias de la capa de dominio, las dependencias se pueden incluir o eliminar desde el archivo ‘pom.xml’. Al agregar una nueva dependencia a este archivo ‘pom.xml’ spring automáticamente se encarga de instalar la dependencia, almacenándola en la carpeta ‘Maven Dependencies’.

### Capa de Persistencia

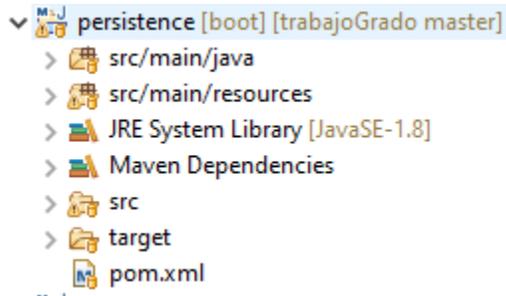


Figura 7.2.11

Autor: Autoría propia.

En la carpeta ‘persistence’ se encuentran 6 paquetes y un archivo, cada paquete cumple con unas tareas específicas para el correcto funcionamiento de la capa de persistencia, y el archivo ‘pom.xml’ es donde se encuentran configuradas las dependencias del proyecto.

- Paquete src/main/java

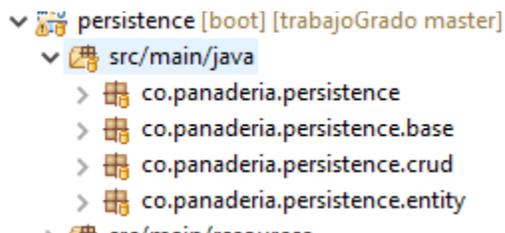


Figura 7.2.12

Autor: Autoría propia.

En el paquete ‘src/main/java’ se encuentra el código fuente, este contiene cuatro paquetes, encargados del tratado de datos hacia la base de datos, permitiendo trabajar con la base de datos relacional por medio de objetos Java.

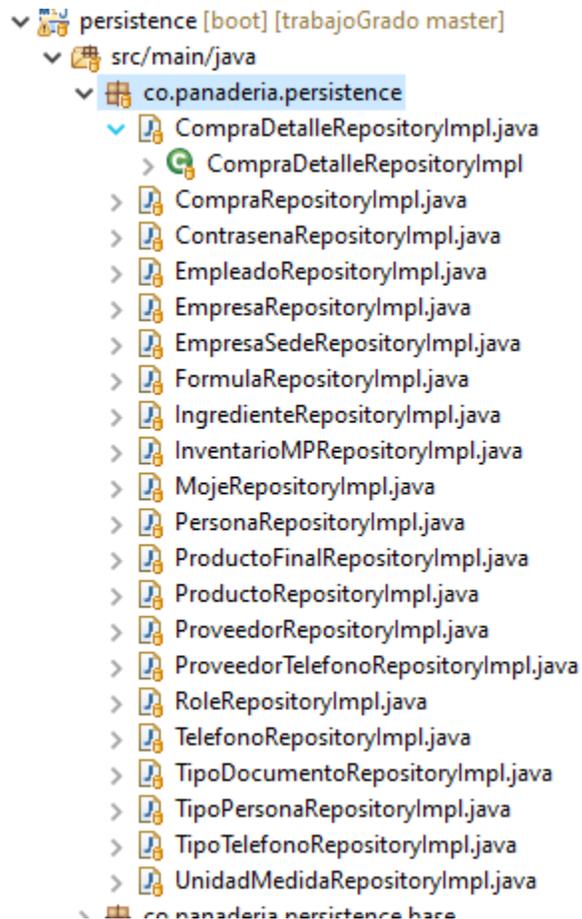


Figura 7.2.13

Autor: Autoría propia.

El primer paquete denominado ‘co.panaderia.persistence’ contiene las clases encargadas de gestionar todas las operaciones de la persistencia contra las tablas de la base de datos, cada clase contiene los métodos e instancias necesarias para realizar las operaciones. Estas clases extienden del repositorio genérico de la capa de dominio; estas son las interfaces mencionadas anteriormente en la capa de dominio del paquete ‘co.panaderia.domain.repository’.

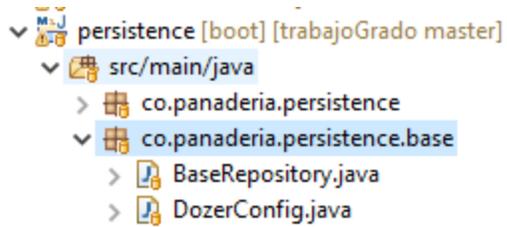


Figura 7.2.14

Autor: Autoría propia.

El segundo paquete ‘co.panaderia.persistence.base’ contiene dos clases, las cuales se encargan del casteo entre los atributos de las entidades de dominio con las entidades del JPA de la base de dato. La primera clase ‘BaseRepository.java’ es una clase genérica que se encarga de hacer la conversión de objetos por medio del framework de Dozer. En la segunda clase ‘DozerConfig.java’, se encuentra un array con los nombres de los archivos de casteo que Dozer necesita para realizar la conversión. Estos archivos son leídos desde el paquete ‘src/main/resources’ el cual se explicará más adelante.

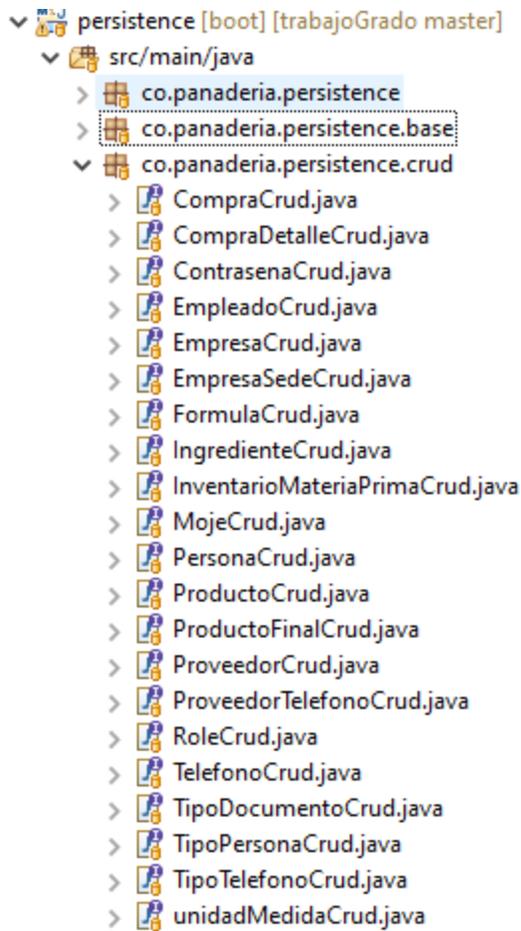


Figura 7.2.15

Autor: Autoría propia.

El siguiente paquete ‘co.panaderia.persistence.crud’ se encuentran las clases tipo interfaces, las cuales tienen la función de realizar las consultas a la base de datos. Es decir, allí están todos los queries y consultas que traen la información de la base de datos.

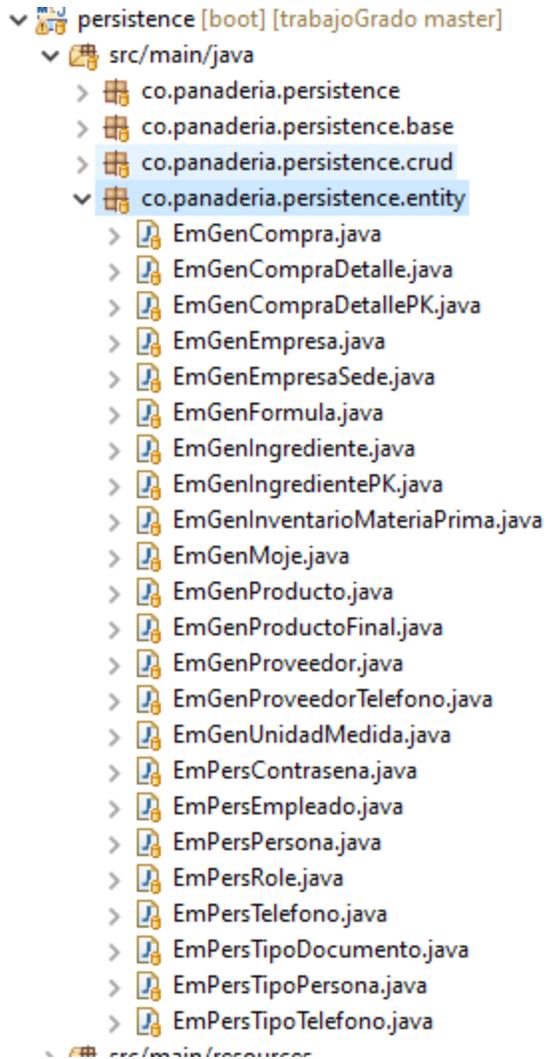


Figura 7.2.16

Autor: Autoría propia.

El cuarto y último paquete ‘co.panaderia.persistence.entity’ contiene las entidades de dominio de JPA, cada clase corresponde a una entidad de dominio, y en ellas está la lógica que realiza la conversión de las tablas de la base de datos a un objeto java, el cual contiene los atributos, relaciones entre tablas y los respectivos getters y setters. Estas clases permiten trabajar con las tablas de base de datos como un objeto java que podemos declarar.

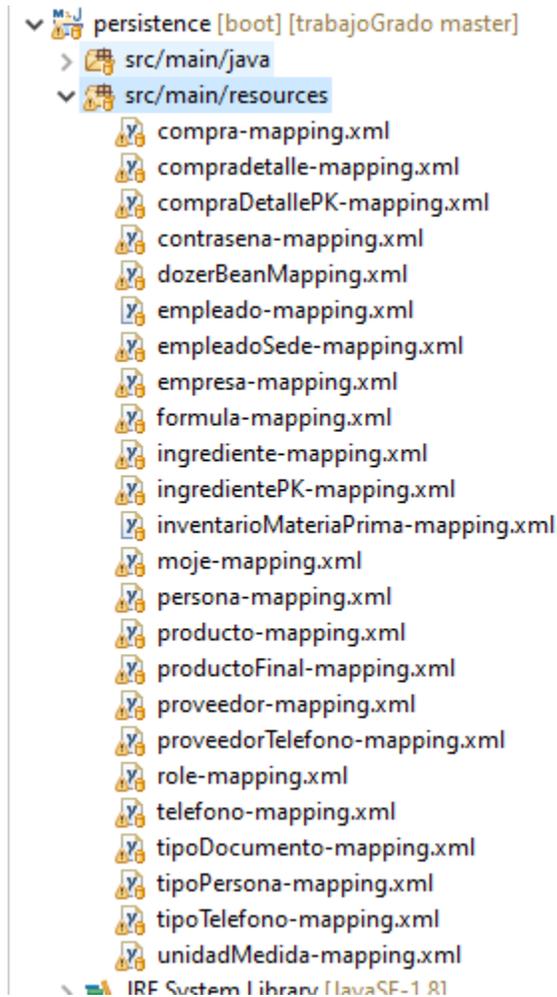
*Paquete src/main/resources*

Figura 7.2.17

Autor: Autoría propia.

En el paquete ‘src/main/resources’ se encuentran los archivos Dozer con extensión .xml, que corresponden a las entidades de dominio para el casteo de los datos. Cada archivo internamente contiene los atributos de las entidades de dominio y los atributos de las entidades de dominio del JPA, donde se hace la referencia de que el atributo de dominio corresponde al mismo atributo de dominio del JAP de la base de datos. En la Figura 7.2.17 se puede ver un ejemplo de ello:

```

1 |<?xml version="1.0" encoding="UTF-8"?>
2 |< mappings xmlns="http://dozer.sourceforge.net" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
3 |   xsi:schemaLocation="http://dozer.sourceforge.net http://dozer.sourceforge.net/schema/beanmapping.xsd">
4 |   < mapping>
5 |     < class-a>co.panaderia.persistence.entity.EmPersPersona</class-a>
6 |     < class-b>co.panaderia.domain.Persona</class-b>
7 |
8 |
9 |     < field>
10 |       < a>idPersona</a>
11 |       < b>idInterno</b>
12 |     </field>
13 |     < field>
14 |       < a>emGenEmpresaSede.idSede</a>
15 |       < b>idSede</b>
16 |     </field>
17 |     < field>
18 |       < a>emPersTipoPersona.idTipoPersona</a>
19 |       < b>idTipoPersona</b>
20 |     </field>
21 |     < field>

```

Figura 7.2.18

Autor: Autoría propia.

Este archivo hace referencia a la entidades de 'Persona', donde se realiza el casteo entre los atributos de las entidades de dominio <class-a> con las entidades de dominio del JPA <class-b>, con ello el sistema entiende a que campo o atributo corresponde.

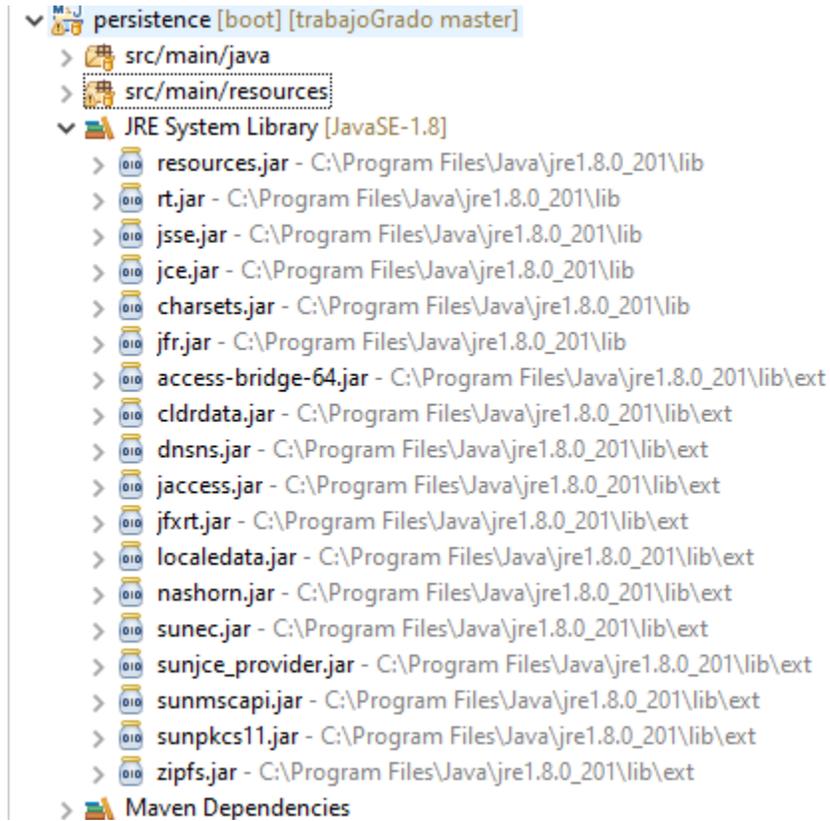
*Paquete JRE System Library*

Figura 7.2.19

Autor: Autoría propia. Librerías del framework Spring.

En este paquete contiene todas las librerías que necesita el sistema para el correcto funcionamiento de la capa de la persistencia.

### Paquete Maven Dependencias

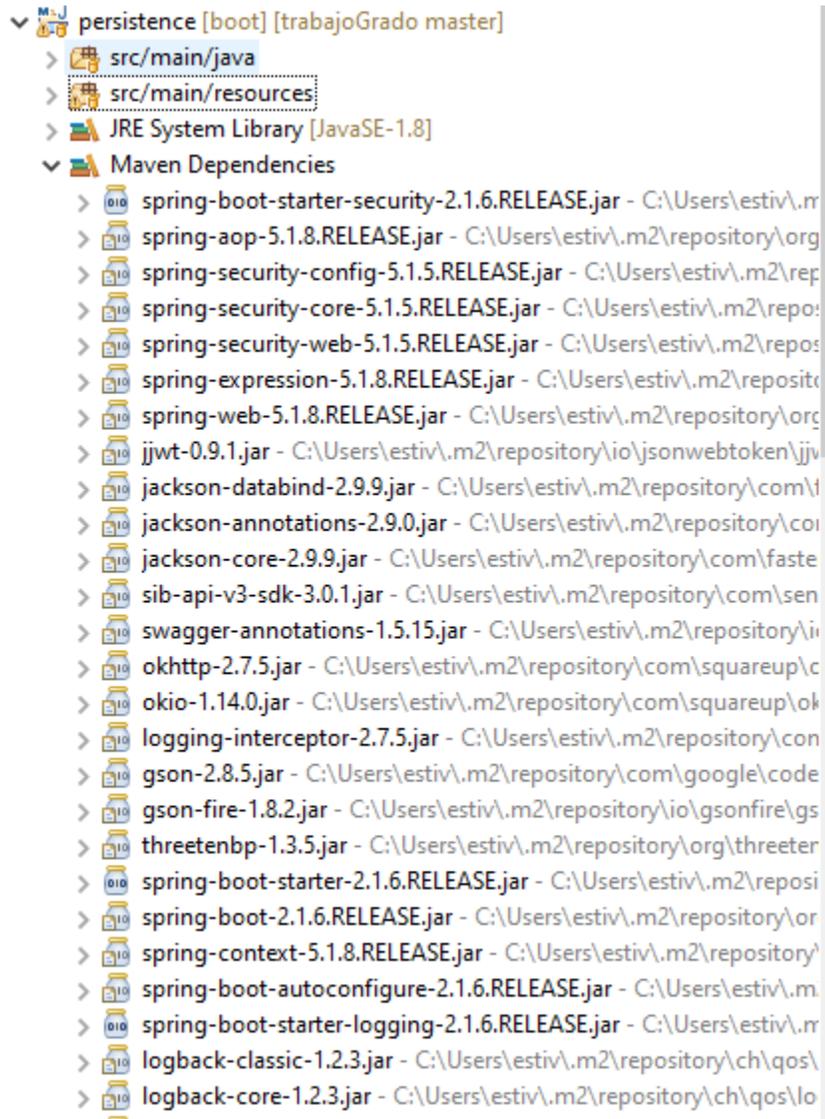


Figura 7.2.20

Autor: Autoría propia. Dependencias del framework Spring.

En esta carpeta se encuentran las dependencias de la capa de la persistencia, las dependencias se pueden incluir o eliminar desde el archivo 'pom.xml'. Al agregar una nueva dependencia a este archivo 'pom.xml' spring automáticamente se encarga de instalar dicha dependencia, almacenándola en la carpeta 'Maven Dependencies'.

## Capa Web

En la capa web se tiene lo siguiente:

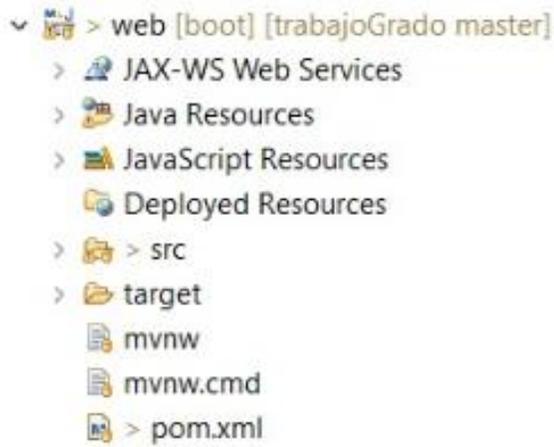


Figura 7.2.21

Autor: Autoría propia.

La capa web está conformada por cinco paquetes y un archivo. El paquete ‘Java Resources’ contiene otros paquetes internamente, lo cuales tienen la lógica de seguridad con JWT y los controladores que exponen los servicios Rest. Y a parte de este paquete se tiene el archivo de configuración de dependencias ‘pom.xml’.

### *Paquete Java Resources*



Figura 7.2.22

Autor: Autoría propia.

Este paquete contiene otras tres carpetas a su vez internamente contienen más carpetas. El paquete ‘Java Resources’ tiene el paquete de ‘src/main/java’ con la lógica de seguridad, los servicios Rest. También el paquete de ‘src/main/resources’ donde se encuentra el archivo de configuración de propiedades y un paquete de librerías, que son necesarias para el correcto funcionamiento del sistema.

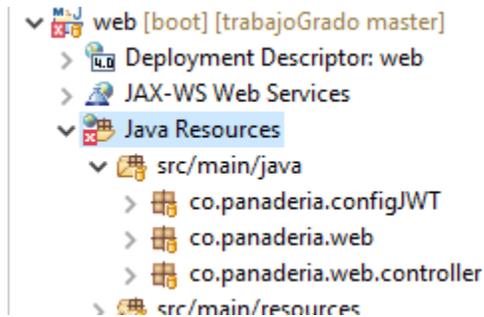


Figura 7.2.23

Autor: Autoría propia.

De manera más detallada, El primer paquete ‘src/main/java’ contiene tres paquetes con la lógica, donde la carpeta ‘co.panaderia.configJWT’ es el encargado de la seguridad, en la Figura 7.2.24 se puede observar las clases que implementaron la seguridad.

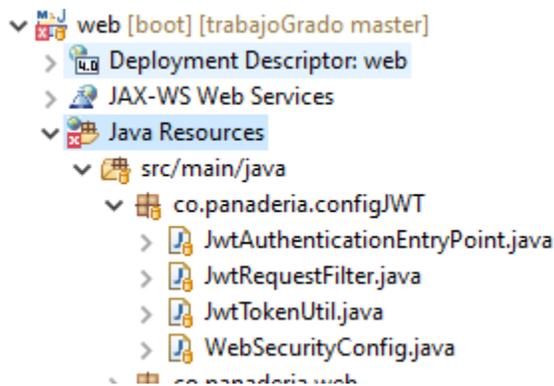


Figura 7.2.24

Autor: Autoría propia.

Este contiene cuatro clases, la primera de ellas ‘JwtAuthenticationEntryPoint.java’ encargada del manejo de excepciones, en caso de errores de autenticación. La segunda clase ‘JwtRequestFilter’ está encargada de la autenticación, es decir valida que el usuario que está navegando en la aplicación se encuentre autenticado, de no estarlo negará el acceso a las funcionalidades del sistema. La tercera clase ‘JwtTokenUtil’ encargada de generar el token con su respectivo cuerpo de información, es decir, la información que viaja internamente en el token. Finalmente, la cuarta clase ‘WebSecurityConfig’ encargada de cargar la encriptación de las contraseñas y validar que sean las mismas que se están ingresando a la aplicación contra las que hay en la base de datos.

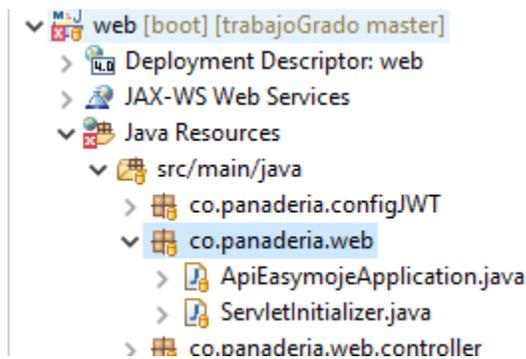


Figura 7.2.25

Autor: Autoría propia.

El paquete de ‘co.panaderia.web’ hay dos archivos java, encargadas del inicio de la aplicación. En la primera clase ‘ApiEasymojeApplication’ contiene el ‘main’ de la aplicación el cuál se encarga de cargar los componentes necesarios para la ejecución de la aplicación. Y en la segunda clase ‘ServletInitializer’ se encarga de hacer posible el inicio de la ejecución de Spring como aplicación web.

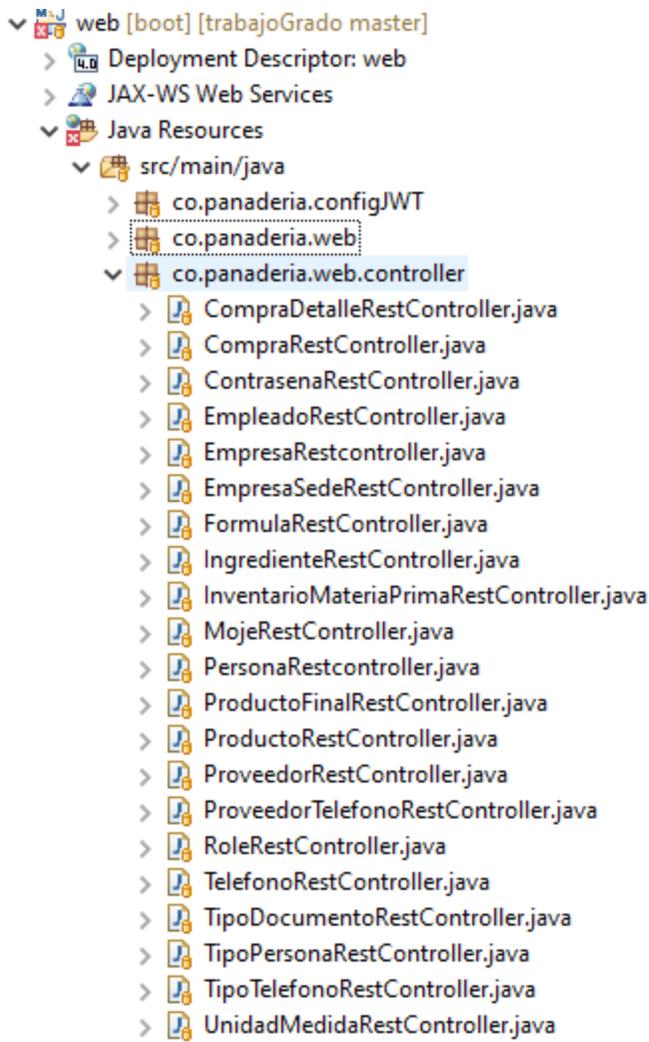


Figura 7.2.26

Autor: Autoría propia.

En la Figura 7.2.26 se observa el paquete de “co.panaderia.web.controller” el cual contiene todos los controladores web de la aplicación. Cada controlador contiene diferentes servicios Rest los cuales son expuestos al frontend para hacer uso de ellos. Estos servicios corresponden a las diferentes funcionalidades del backend entre ellos los servicios del CRUD (crear, leer, actualizar y eliminar).

*Paquete Java Script Resources*

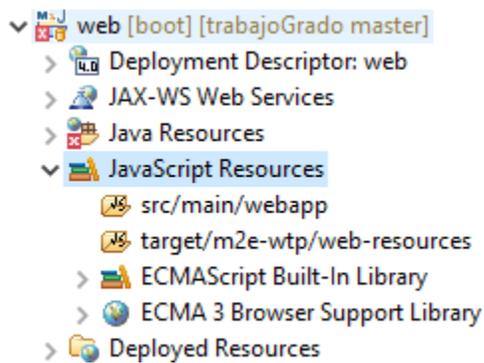


Figura 7.2.26

Autor: Autoría propia.

En esencia el paquete ‘Java Script Resources’ contiene archivos que contienen librerías que se encargan del correcto funcionamiento de la capa web.

Finalmente tenemos archivo ‘application.properties’(ver Figura 7.2.27) encargado de la configuración y conexión a la base de datos de MySQL, allí se declara la contraseña, usuario y ruta de la base de datos.

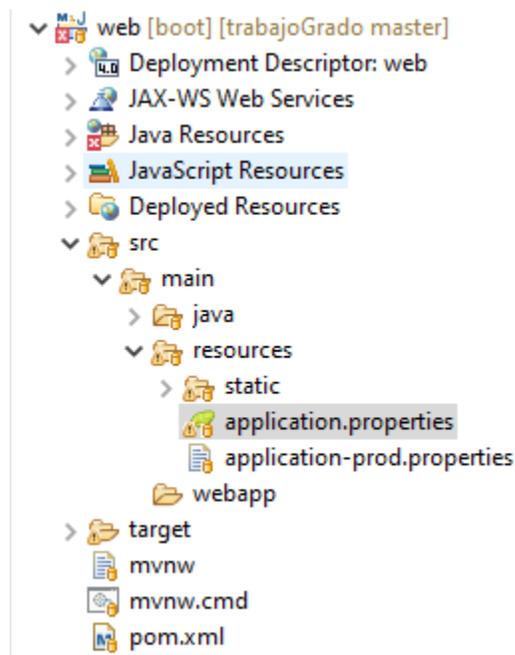


Figura 7.2.27

Autor: Autoría propia.

### 7.3 Diseño del Frontend

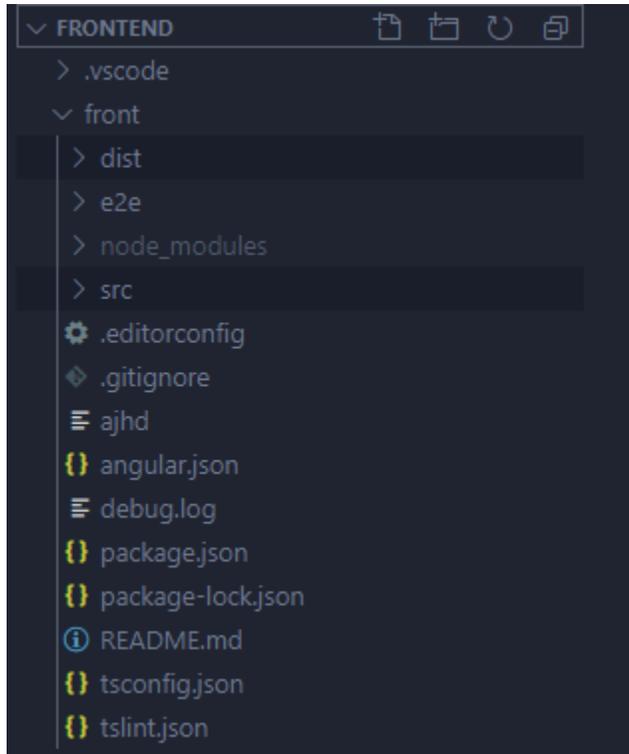


Figura 7.3.1

Autor: Autoría propia. Frontend.

Dentro de la carpeta de 'front' se encuentra todo el contenido de la presentación. Este contenido está repartido en varias subcarpetas y archivos, donde cada uno juega un papel fundamental para el correcto funcionamiento de la presentación.

El paquete 'dist' se puede decir que es una carpeta que contiene en varios archivos la compresión de todo el frontend, con la versión de tu aplicación. También, está el paquete 'e2e', este tiene relación a los desarrollos de pruebas. Luego, está el paquete de 'node\_modules' es donde están componentes instalados para facilitar el desarrollo, estos componentes son como las librerías y dependencias del front. El paquete 'src' es donde se encuentra todo el desarrollo del

código fuente de la aplicación del front, es decir, todos los componentes, vistas, servicios, enrutadores que hacen posible que la capa de presentación funcione.

Cómo el código fuente desarrollado se encuentra en la carpeta de 'src', será el enfoque principal del proyecto de frontend.

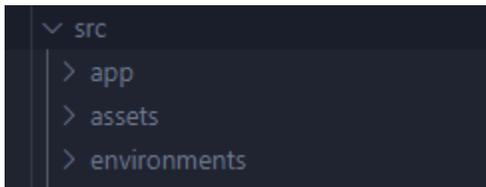


Figura 7.3.2

Autor: Autoría propia.

El paquete de 'src' se conforma en tres subcarpetas donde la carpeta de 'app' tiene el código fuente, la carpeta 'assets' contiene las imágenes que necesita la aplicación en las vistas. Las imágenes fueron tomadas de 'Pixaby' un repositorio de imágenes gratuitas. Finalmente, la carpeta 'environments' que contiene el endPoint de conexión al backend. Se hará más énfasis en la carpeta de 'app' ya que esta contiene el desarrollo.

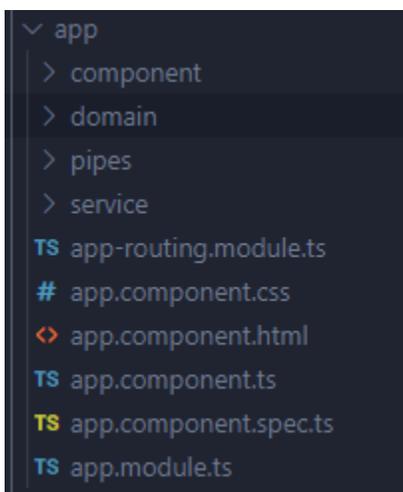


Figura 7.3.3

Autor: Autoría propia.

El paquete 'app' contiene cuatro carpetas y seis archivos, los cuales se darán detalle a continuación. El archivo 'app-routing.module.ts' contiene todas las rutas o paths de los componentes de la aplicación, y se encarga del enrutamiento de los estos componentes, cada uno de estos componentes tiene un patch asignado. El siguiente archivo 'app.component.css', es un documento para estilizar el archivo de 'app.component.html'. Luego está el archivo 'app.component.html', este es la vista que contiene la estructura de la página e internamente contiene las etiquetas del navbar, el body y el footer. En el archivo 'app.component.ts', se tiene la lógica para el funcionamiento de 'app.component.html'. También está el archivo 'app.component.spec.ts' el cual se usa para realizar test o pruebas al código realizado. Finalmente, en el archivo 'app.module.ts' es donde se encuentran declarados todos los componentes de la aplicación y donde se realizan las importaciones de estos.

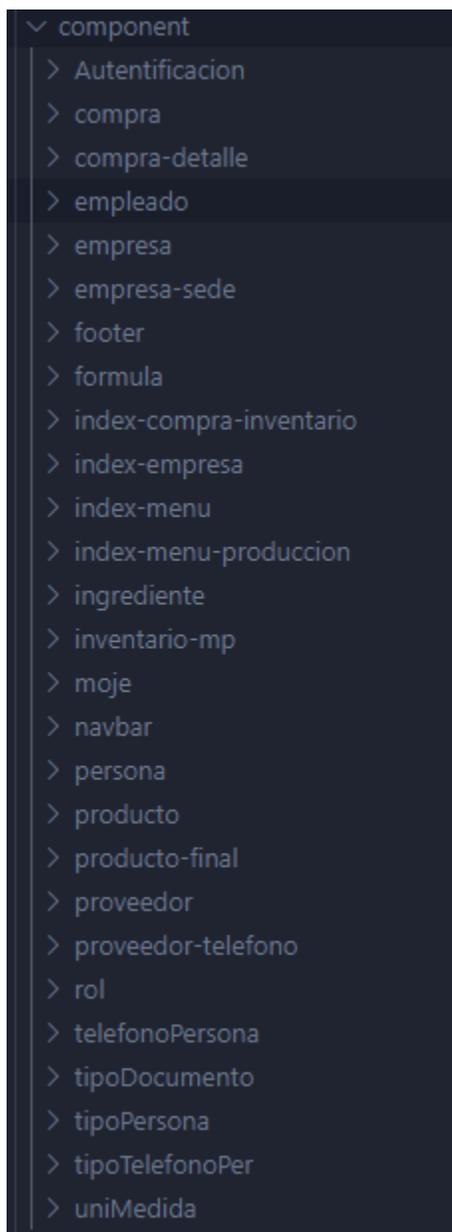
*Paquete con los componentes*

Figura 7.3.4

Autor: Autoría propia. Componentes del frontend.

El paquete 'component' contiene internamente los componentes de la aplicación, cada componente tiene tareas específicas de acuerdo a su nombre. Hay componentes que internamente tienen más componentes, cada componente está compuesto por un archivo.css de estilo, un

archivo.html para la vista de la página, un archivo.ts con la lógica que permite funcionar la vista y finalmente un archivo.spec.ts para probar el código desarrollado. Cada componente se desarrolló usando Angular7, Bootstrap 4, TypeScript, y Html 5.

Los componentes hacen uso de servicios que permiten traer la información que requiere mostrar la vista e insertar información que requiera el usuario. Estos servicios se explicarán más adelante. Además, algunos componentes hacen uso de imágenes para que la aplicación tenga una buena experiencia de usuario al navegar en la aplicación.

También, hay componentes que hacen uso de frameworks que permitieron un desarrollo más sencillo de la presentación, uno de ellos es la librería de 'jwt-decode' que se usó para la seguridad.

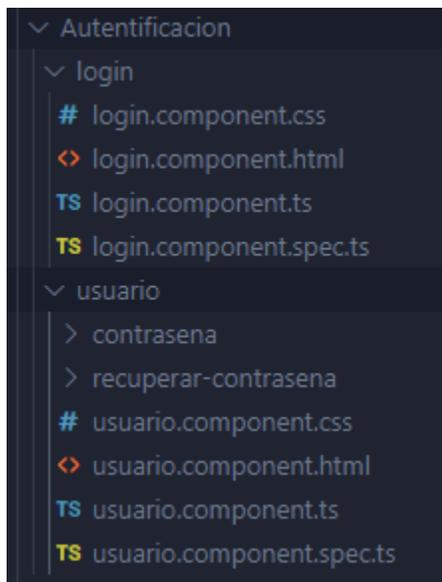


Figura 7.3.5

Autor: Autoría propia. Componente Autenticación.

El componente 'Autenticación' contiene otros dos componentes, donde el componente 'login' tiene las vistas y la lógica del login para el ingreso de la aplicación. El otro componente

‘Usuario’ contiene todo lo referente a perfilación de usuario, como consulta de perfil de empelado, actualizar contraseña y recuperar contraseña.

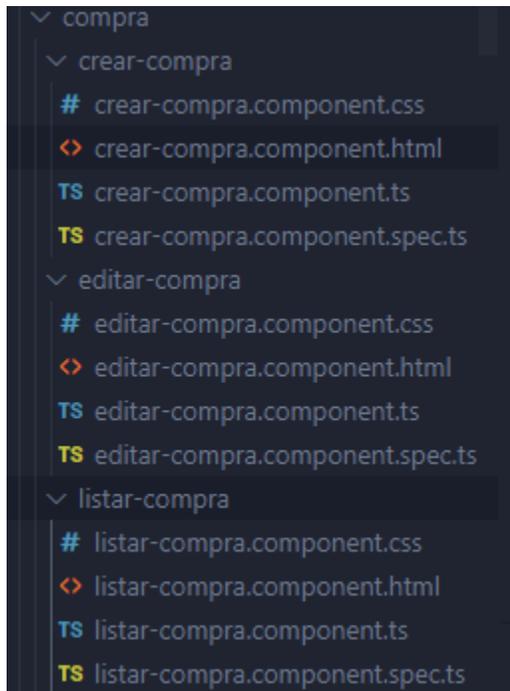


Figura 7.3.6

Autor: Autoría propia. Componente compra.

El componente ‘compra’ contiene la lógica y las vistas que permiten ingresar, actualizar, eliminar e ingresar las compras que nutren el inventario de materias primas.

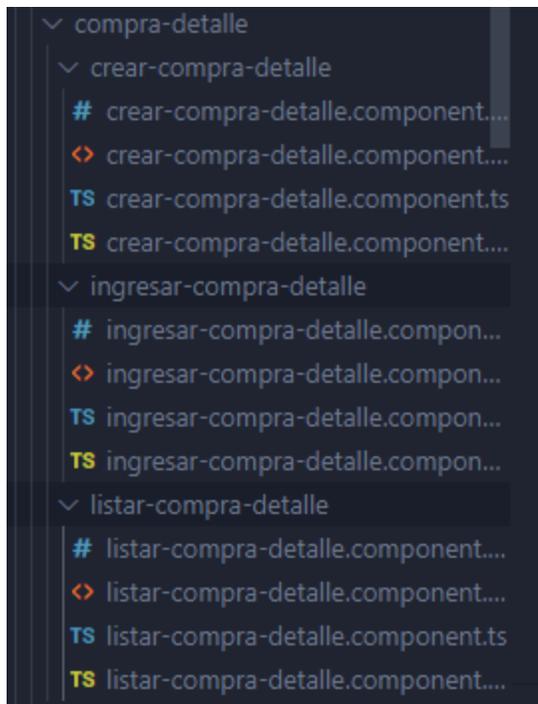


Figura 7.3.7

Autor: Autoría propia. Componente compra-detalle.

El componente 'compra-detalle' contiene la lógica y las vistas que permiten ingresar, actualizar, eliminar e ingresar el detalle de una compra realizada, este componente depende del componente 'compra'.

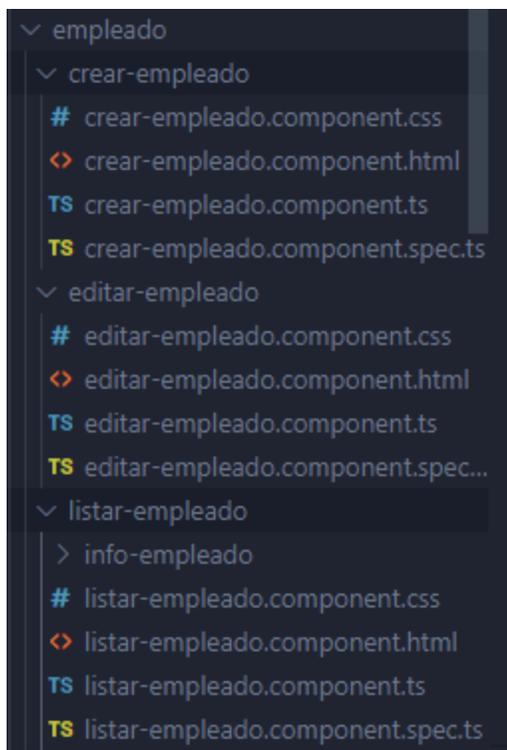


Figura 7.3.8

Autor: Autoría propia. Componente empleado.

El componente 'empleado' contiene la lógica para crear, editar actualizar, consultar y eliminar un empleado. Además, para asignar o eliminar las credenciales a un empleado.

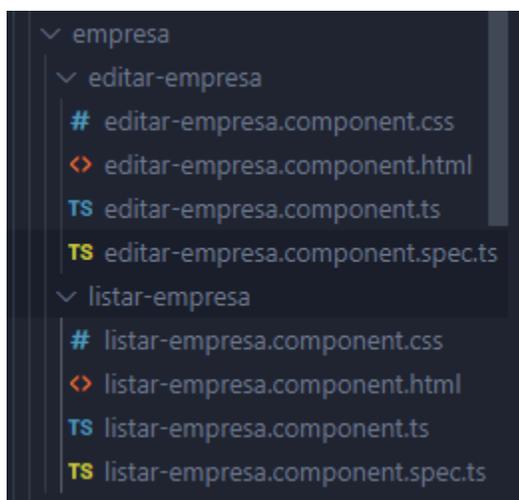


Figura 7.3.9

Autor: Autoría propia. Componente empresa.

Este componente 'empresa' contiene la lógica y las vista para ver y editar la información una empresa.

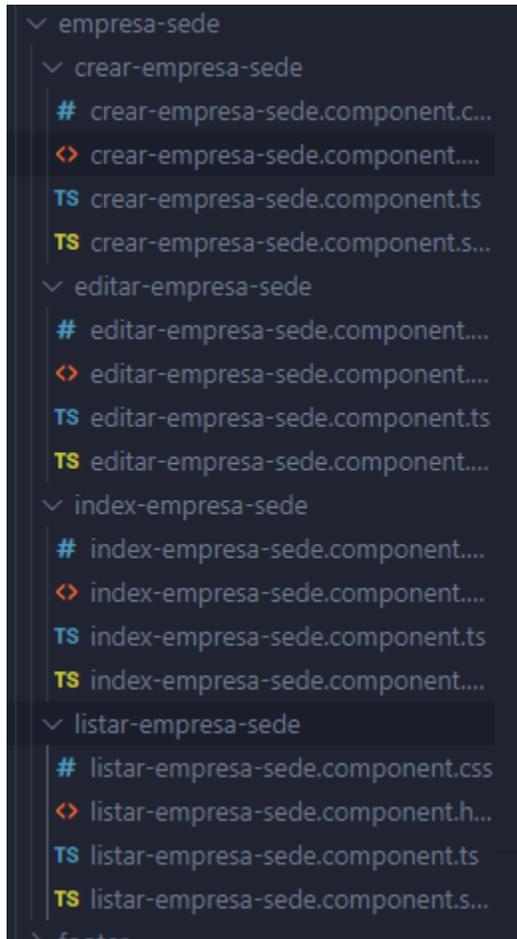


Figura 7.3.10

Autor: Autoría propia. Componente sede.

Este componente permitió ver las sedes de una empresa, y contiene la lógica para editar y actualizar la información de la sede. Además, permite ingresar al menú principal de la aplicación, separando los datos por sede.

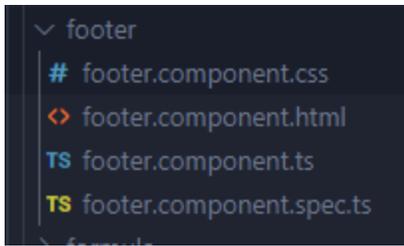


Figura 7.3.11

Autor: Autoría propia. Componente footer.

Este componente 'footer' es el pie de página de cada vista de la aplicación, este componente se usa en todas las vistas.

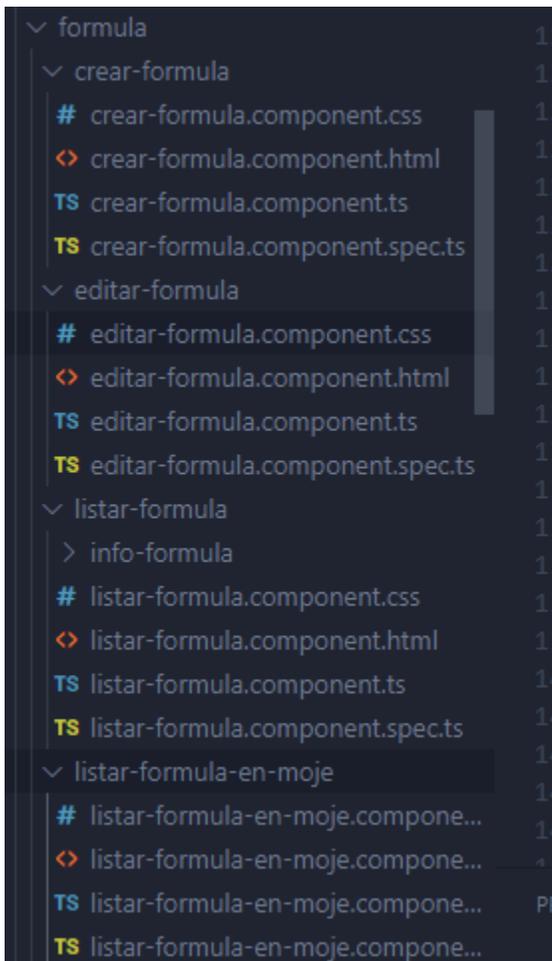


Figura 7.3.12

Autor: Autoría propia. Componente formula.

El componente ‘formula’, contiene internamente más componentes, donde cumplen con la función de crear una formula, actualizar y ver la información de una fórmula.



Figura 7.3.13

Autor: Autoría propia. Componente index-compra-inventario.

Este componente ‘index-compra-inventario’ contiene la lógica y la vista principal del módulo de compras e inventario, se realizó para poder acceder a las opciones de compras, inventario y proveedores.



Figura 7.3.14

Autor: Autoría propia. Componente index-empresa.

El componente ‘index-empresa’ contiene la vista principal de una empresa y el ingreso a las sedes de una empresa. Desde allí se permite editar y ver la información de la empresa.

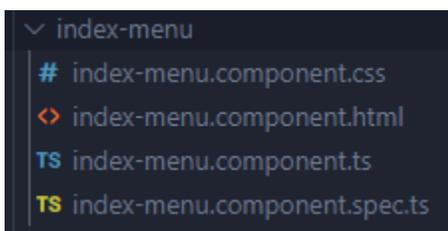


Figura 7.3.15

Autor: Autoría propia. Componente index-menu.

El componente 'index-menu' tiene el papel del menú principal de la aplicación, este se realizo para acceder a las funcionalidades de la aplicación cómo lo es el módulo de producción, de talento humano y de compras e inventario.

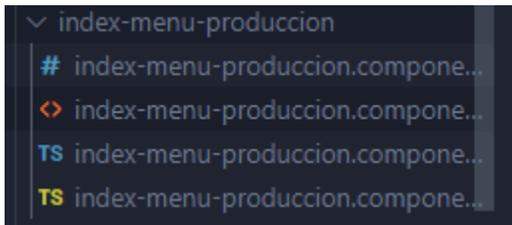


Figura 7.3.16

Autor: Autoría propia. Componente index-produccion.

Este componente 'index-produccion' contiene la vista y la lógica para seleccionar las diferentes funcionalidades del módulo de producción.

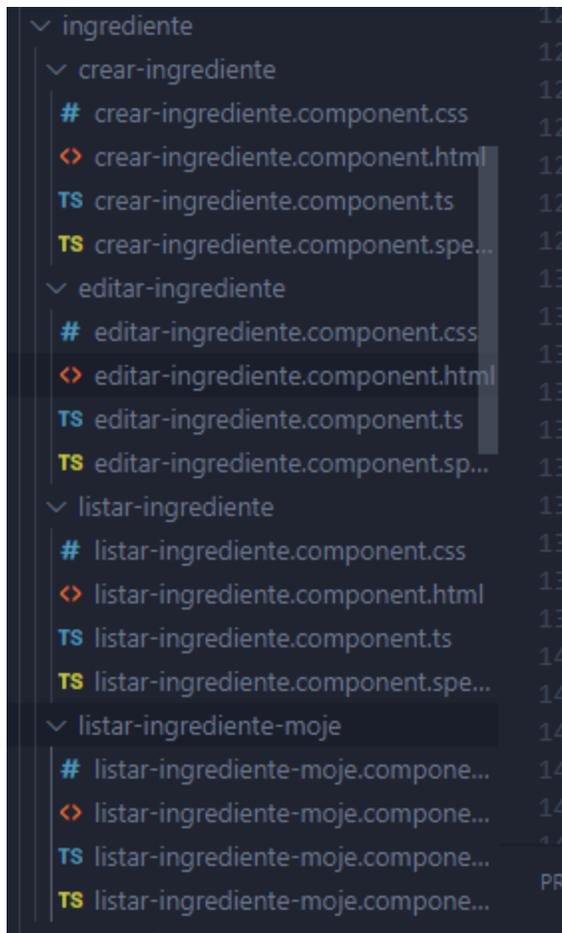


Figura 7.3.17

Autor: Autoría propia. Componente ingrediente.

Este componente 'ingrediente' se encarga de permitir añadir ingredientes a una fórmula, también de crear ingrediente, ver ingredientes de una fórmula, y otras funcionalidades más.

▼ inventario-mp	122
▼ crear-inventario	123
# crear-inventario.component.css	124
◊ crear-inventario.component.html	125
TS crear-inventario.component.ts	126
TS crear-inventario.component.spec.ts	127
▼ editar-inventario	128
# editar-inventario.component.css	129
◊ editar-inventario.component.html	130
TS editar-inventario.component.ts	131
TS editar-inventario.component.spec...	132
▼ listar-inv-bajo	133
> info-inv	134
# listar-inv-bajo.component.css	135
◊ listar-inv-bajo.component.html	136
TS listar-inv-bajo.component.ts	137
TS listar-inv-bajo.component.spec.ts	138
▼ listar-inventario	139
# listar-inventario.component.css	140
◊ listar-inventario.component.html	141
TS listar-inventario.component.ts	142
TS listar-inventario.component.spec.ts	PROBLEMS

Figura 7.3.18

Autor: Autoría propia. Componente inventario-mp.

Este componente ‘inventario-mp’ contiene toda la lógica para ver el inventario de la sede de la panadería, de ver los inventarios de bajos de la sede de la panadería de los insumos que están agotados, además, permite crear y actualizar materias primas. También, contiene la lógica para notificar por medio de correo electrónico a los administradores de la sede los insumos agotados y el inventario de la sede en dos tiempos, uno al inicio de jornada laboral y el otro en la finalización de horario laboral.

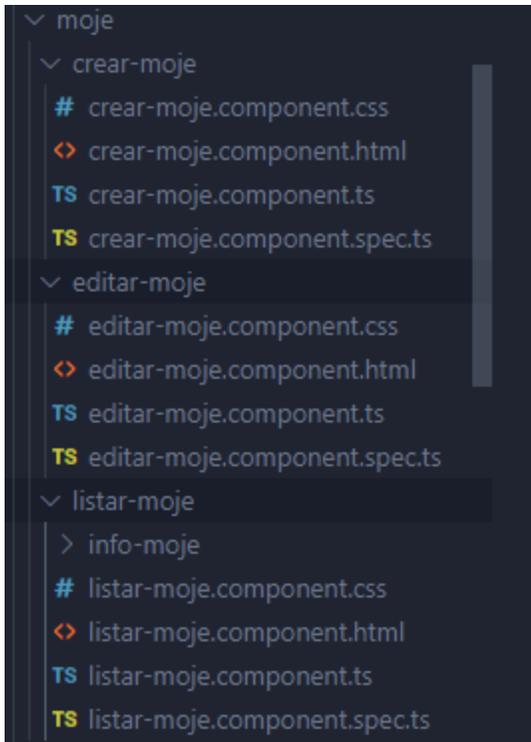


Figura 7.3.19

Autor: Autoría propia. Componente moje.

Este componente, se hizo para permitir al usuario crear mojes de los productos, además de crear el componente contiene la lógica para visualizar los mojes realizados con su información detallada.

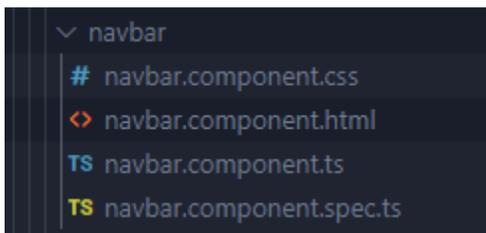


Figura 7.3.20

Autor: Autoría propia. Componente navbar.

El componente 'navbar' es la barra de menú de encabezado de la aplicación, este componente es general para todas las vistas, y contiene la lógica para acceder de manera rápida a otros módulos de la aplicación.

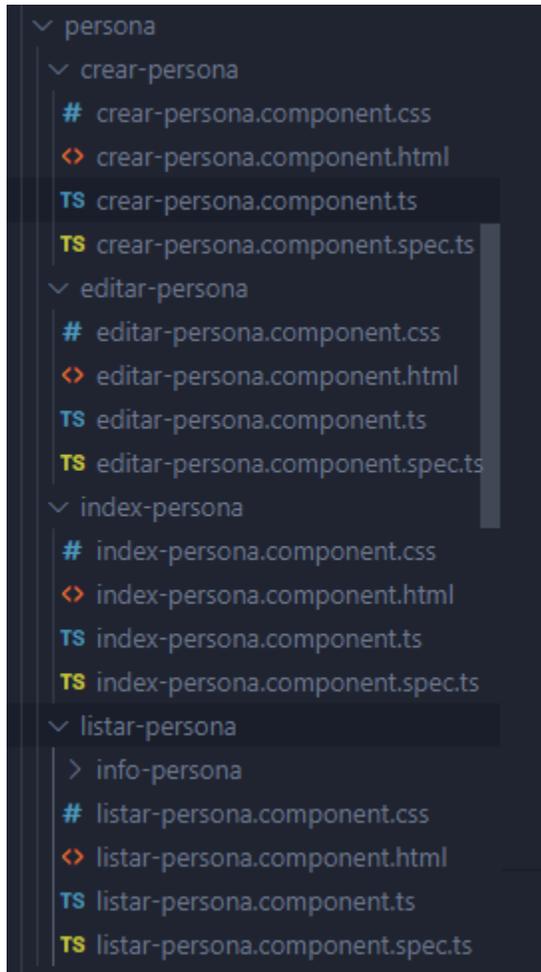


Figura 7.3.21

Autor: Autoría propia. Componente persona.

Este componente 'persona' contiene toda la lógica que permiten al software ingresar personas a una sede, actualizar la información ingresada, eliminar personas, añadir teléfonos y visualizar la información de detallada de cada persona.

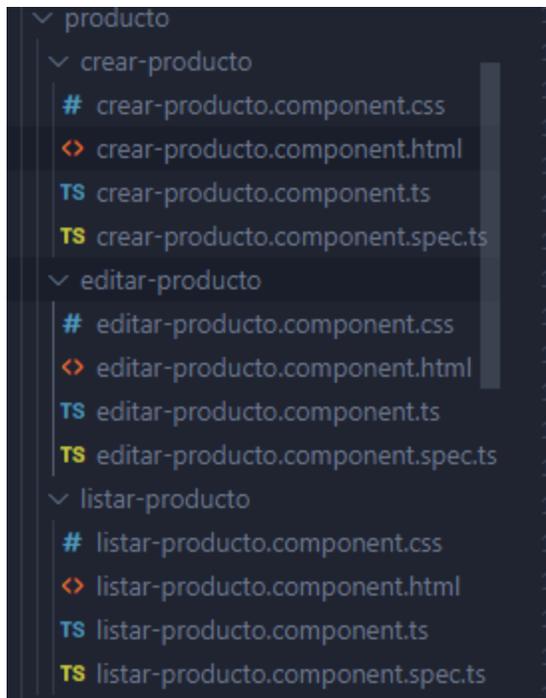


Figura 7.3.22

Autor: Autoría propia. Componente producto.

Este componente 'producto' se realizó para permitir crear un producto de panadería, el cual es usado por otros componentes de la aplicación como formulas y mojes. Además, tiene la capacidad de actualizar y observar la información de los productos creados.

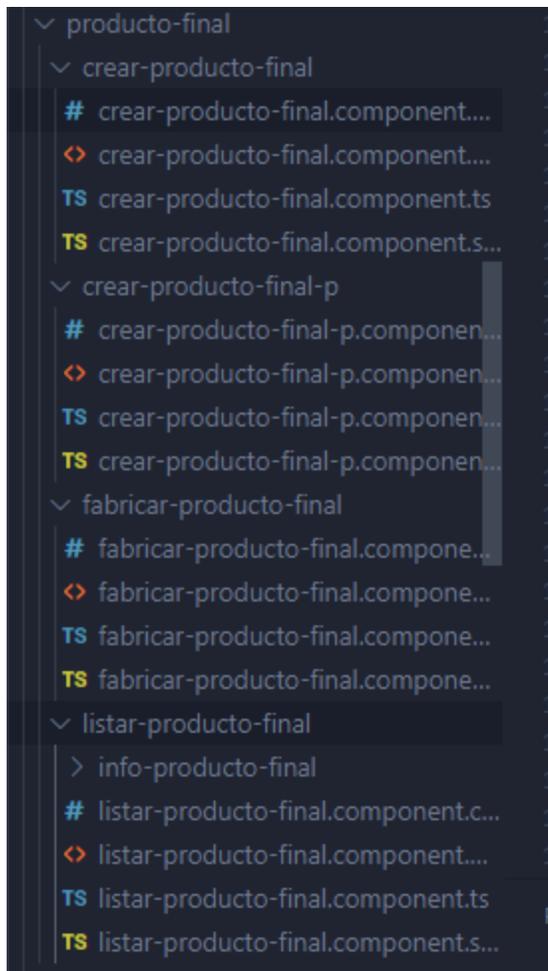


Figura 7.3.23

Autor: Autoría propia. Componente producto-final.

El componente de 'producto-final' se realizó para permitir crear un producto terminado, este depende de la realización de un moje el cual es usado para generar la producción final. Además, este contiene la lógica para realizar otras tareas como visualización de información de los mojes y productos finales.



Figura 7.3.24

Autor: Autoría propia. Componentes proveedor y teléfono-proveedor.

Estos dos componentes contienen la lógica que permitió ingresar proveedores de las materias primas que se usan al ingresar compras. Además, contiene las vistas de visualización de la información de los proveedores.

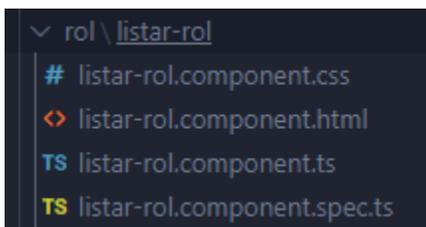


Figura 7.3.25

Autor: Autoría propia. Componente rol.

El componente 'rol' se hizo para permitir al usuario visualizar los roles que tiene el sistema. Estos roles están parametrizados desde la base de datos, la aplicación no permite ingresarlos.

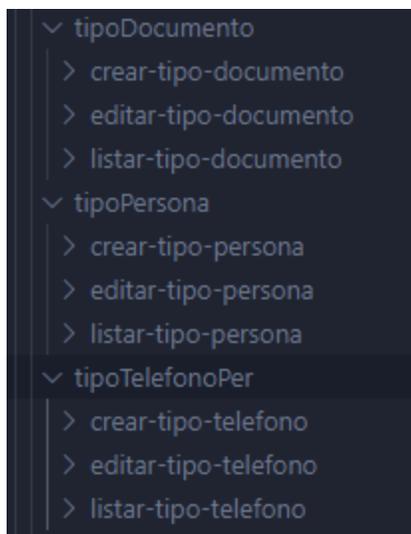


Figura 7.3.26

Autor: Autoría propia.

En los componentes que se pueden observar en la Figura 7.3.26 contiene la lógica para ingresar editar, actualizar y observar la información de los tipos de documentos, tipos de teléfonos, y tipos de persona que se le pueden asignar a una persona.

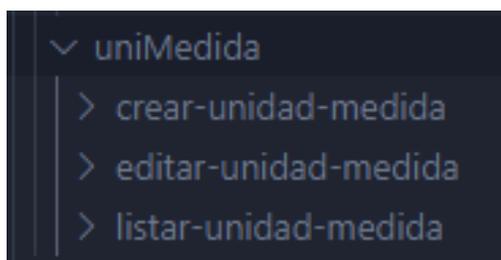


Figura 7.3.27

Autor: Autoría propia. Componente uniMedida.

Finalmente, se tiene el componente 'uniMedidad', este componente se usó para las unidades del software.

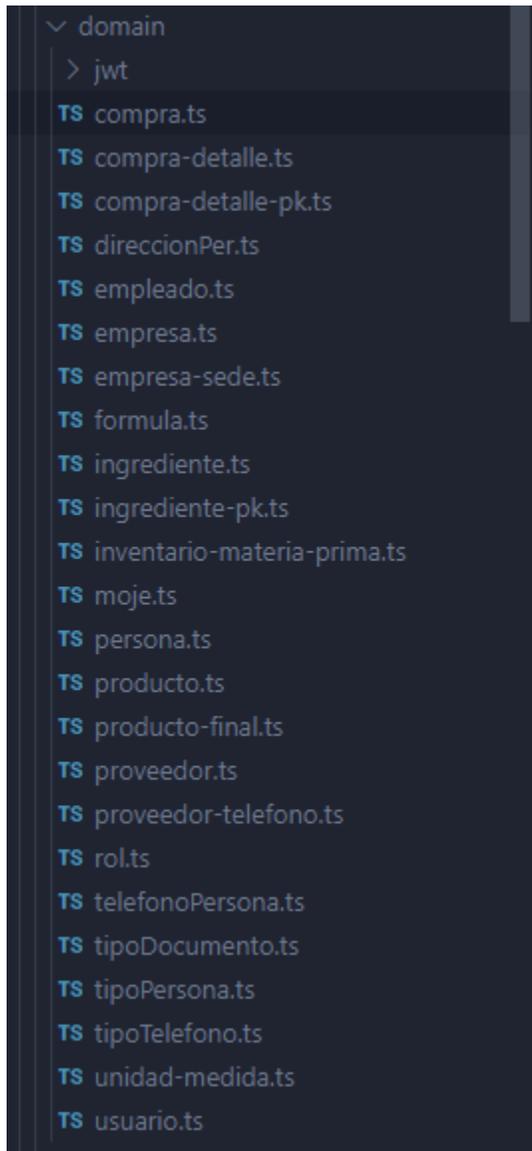
*Carpeta de dominio.*

Figura 7.3.28

Autor: Autoría propia. Dominio del frontend.

En la carpeta 'domain' se encuentran todos los dominios que se utilizaron en el frontend. Cada archivo hace referencia a una clase de dominio que contiene los atributos que son usados para obtener los datos del backend y así mismo, para llevar datos desde el frontend al backend. Cada atributo es del mismo tipo de dato que las entidades del backend.

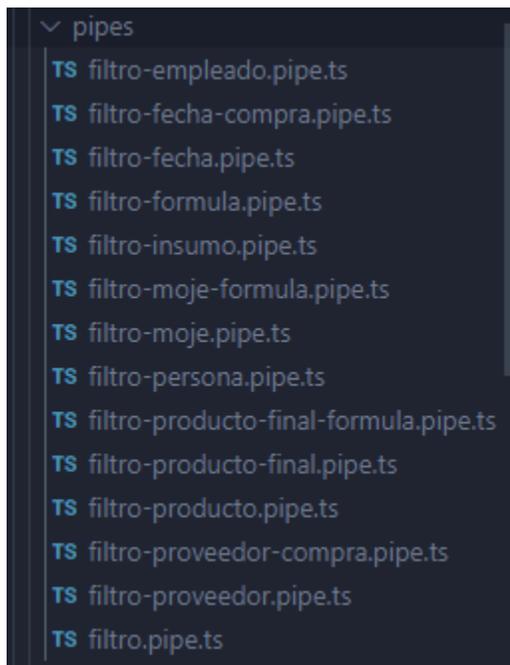
*Carpeta de pipes*

Figura 7.3.29

Autor: Autoría propia.

En la carpeta 'pipes', se encuentran otros archivos que se usaron para crear la lógica para los filtros de búsqueda. Estos filtros no se usaron en todas las vistas, sólo en las que se requería.

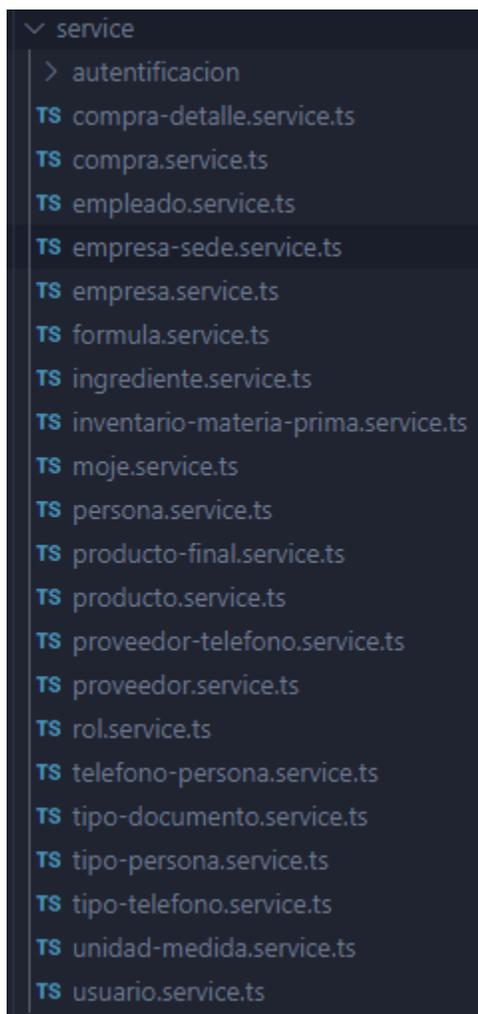
*Carpeta service*

Figura 7.3.30

Autor: Autoría propia.

En la carpeta 'service' se encuentran todo los servicios que permiten la conexión de los servicios REST del backend, es decir, cada servicio contiene los endPoint para consumir los servicios Rest que fueron expuestos por el backend. Estos consumos se hicieron por medio de solicitudes HTTP. Esta carpeta contiene una carpeta intenta llamada 'autenticacion', en ella se encuentran los servicios para la seguridad del software de la presentación.

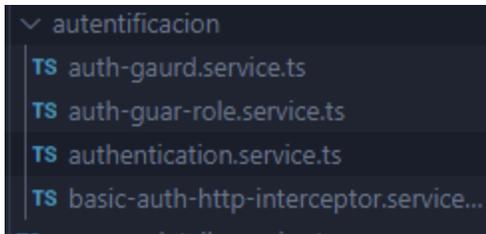


Figura 7.3.31

Autor: Autoría propia.

Estos cuatro servicios se encargan de toda la seguridad del frontend, para ello se usó la librería de 'jwt-decode'. Cuando se realiza alguna petición desde la vista, se solicita un token por parte del backend, el cual es entregado al frontend, donde, por medio de la librería mencionada anteriormente se puede decodificar dicho token y hacer uso de él para beneficios propios.

Desde el token se extrae información como el tipo de rol del usuario que se encuentra navegando, permitiendo controlar y segregar funciones en la aplicación. La clase 'AuthGaurdService.ts' protege las rutas de cada componente para que no sean accedidas sin haber una autenticación previa. Luego, se encuentra el archivo 'AuthGuarRoleService.ts' el cual se realizó para la segregación de funciones en la aplicación, es decir, este servicio permitió controlar las funciones de la aplicación dependiendo del tipo de rol del empleado, este servicio protege también las rutas de acceso de los componentes. El siguiente archivo 'AuthenticationService.ts' se hizo para validar el ingreso del usuario a la aplicación, este solicita las credenciales como usuario y contraseña las cuales van al backend y son validadas contra la base de datos, devolviendo el backend un token con información encriptada en ella. En caso de encontrar error en las credenciales, el servicio no permite el acceso a la aplicación. Por último, se tiene el archivo 'BasicAuthHttpInterceptorService.ts', este se desarrolló para obtener la sesión del usuario, es decir, con este servicio se valida que el usuario que esté navegando en la aplicación, se encuentre autenticado, de lo contrario caducará la sesión.



Las tablas se nombraron de una manera práctica, con el fin de identificarlas de una manera fácil y rápida. Las tablas generales que representan al negocio empiezan por los caracteres 'em\_gen', mientras que las tablas que hacen representación a todo lo relacionado con personas empiezan con los caracteres 'em\_pers'.

Todas las tablas a excepción de la tabla 'em\_gen\_empresa' tienen relación con la tabla de 'em\_gen\_empresa\_sede' ya que la aplicación se hizo para que funcionara separando toda la información por sedes. Todas las tablas tienen por los menos una relación con otra tabla.

A continuación, se mostrará a detalle las tablas, las tablas son las siguientes:

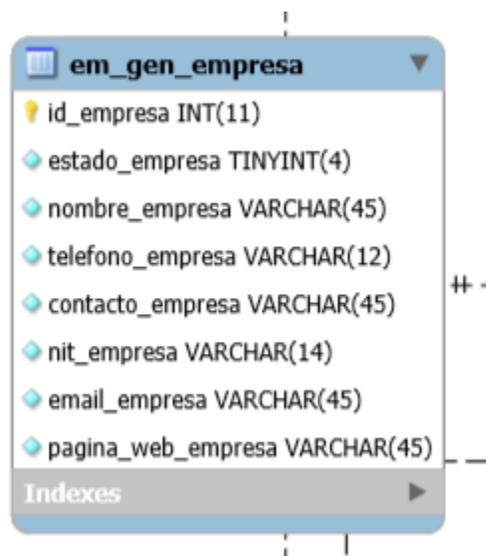


Figura 7.4.2

Autor: Autoría propia, herramienta usada: MySQL Workbench.

La tabla 'em\_gen\_empresa' se hizo para almacenar la información de las empresas. La columna o campo 'id\_empresa' es el identificador único para cada registro, los demás son los atributos de la tabla.

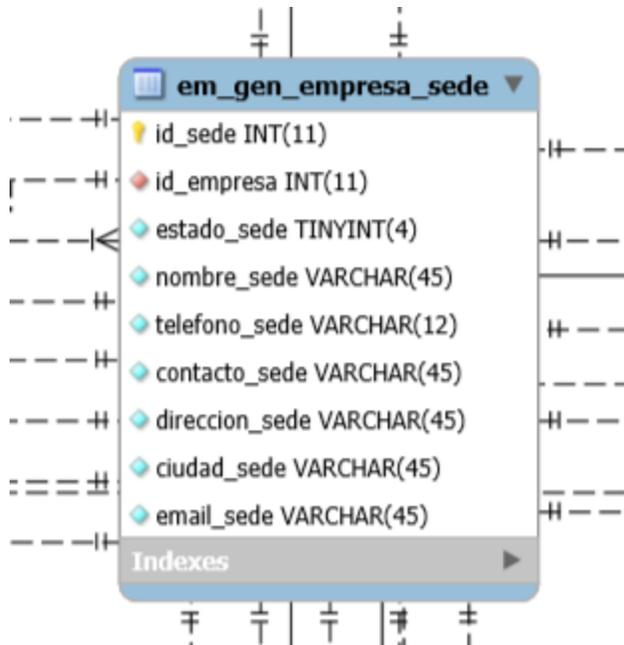


Figura 7.4.3

Autor: Autoría propia, herramienta usada: MySQL Workbench..

La tabla 'em\_gen\_empresa\_sede' se hizo para almacenar la información de las sedes de una empresa. La columna o campo 'id\_sede' es el identificador único para cada registro. Hay otro campo 'id\_empresa' el cual es la llave foránea que referencia una relación con la tabla de 'em\_gen\_empresa', los demás campos son atributos de la tabla.

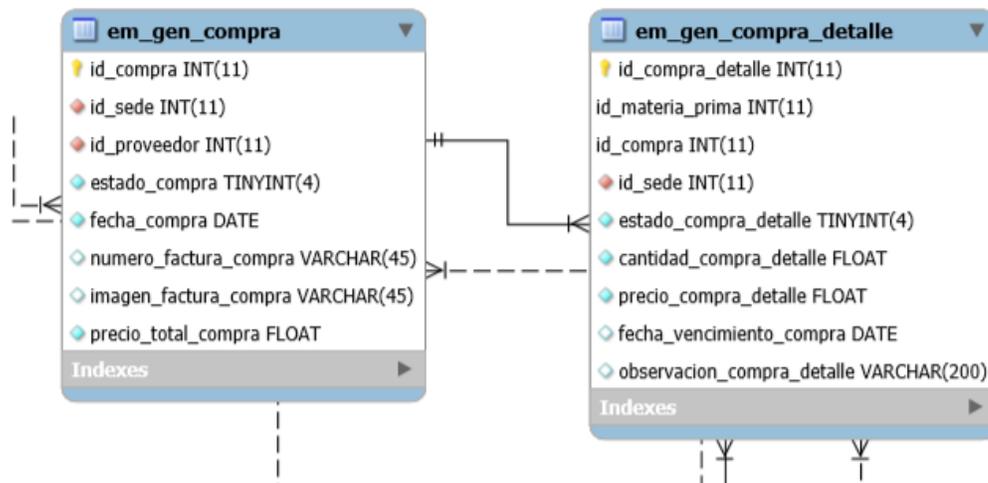


Figura 7.4.4

Autor: Autoría propia, herramienta usada: MySQL Workbench.

Las tablas ‘em\_gen\_compra’ y ‘em\_gen\_compra\_detalla’ se hicieron para almacenar la información de las compras de insumos para la panadería, donde la segunda tabla es la que almacena la información del detalle de una compra. La primera tabla, tiene relación con las tablas de ‘em\_gen\_empresa\_sede’ y ‘em\_gen\_proveedores’. La segunda tabla, tiene relación con las tablas de ‘em\_gen\_compra’, ‘em\_gen\_empresa\_sede’ y ‘em\_gen\_materia\_prima’.

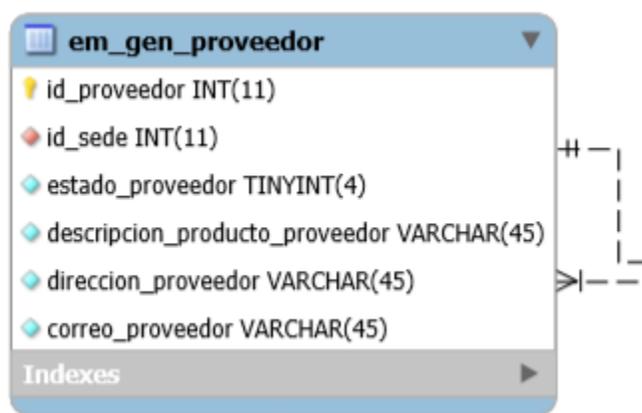


Figura 7.4.5

Autor: Autoría propia, herramienta usada: MySQL Workbench.

La tabla ‘em\_gen\_proveedor’ se hizo para almacenar la información de un proveedor de materia prima. Esta tabla tiene relación con la tabla de ‘em\_gen\_empresa\_sede’.

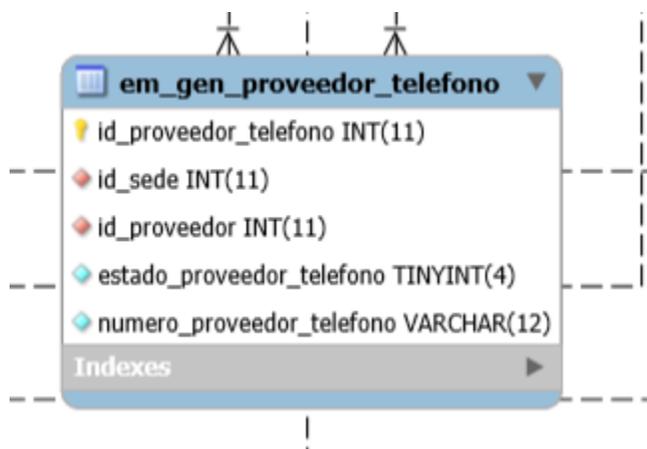


Figura 7.4.6

Autor: Autoría propia, herramienta usada: MySQL Workbench.

La tabla 'em\_gen\_proveedor\_telefono' se realizó para guardar la información de los teléfonos de contacto de un proveedor de una sede. Esta tabla tiene dos claves foráneas, lo cual indica que tiene relación con las tablas de 'em\_gen\_empresa\_sede' y 'em\_gen\_proveedor'.

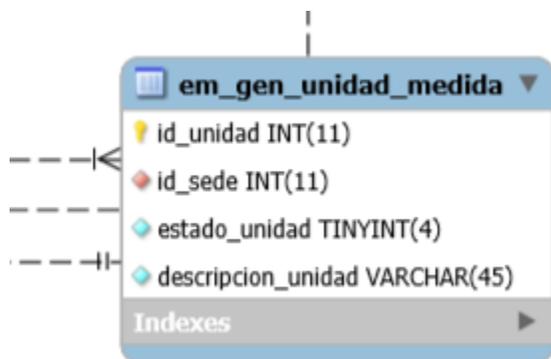


Figura 7.4.7

Autor: Autoría propia, herramienta usada: MySQL Workbench.

La tabla de 'em\_gen\_unidad\_medida' se construyó con el fin de almacenar las unidades de medida que tienen las materias primas en el inventario. Esta tabla tiene una clave foránea, lo cual indica que tiene relación con la tabla de 'em\_gen\_empresa\_sede'.

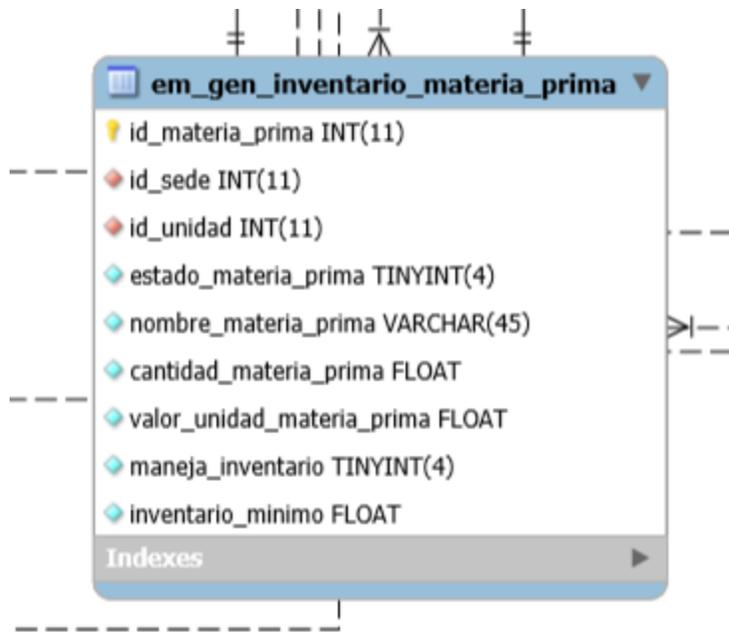


Figura 7.4.8

Autor: Autoría propia, herramienta usada: MySQL Workbench.

La tabla de 'em\_gen\_materia\_prima' se realizó para almacenar la información de las materias primas, es decir, es la tabla de inventario de insumos. Esta tabla a parte de la tabla de 'em\_gen\_empresa\_sede', también tiene relación con la tabla de 'em\_gen\_unidad\_medida'.



Figura 7.4.9

Autor: Autoría propia, herramienta usada: MySQL Workbench.

La tabla de 'em\_gen\_producto' se hizo con el fin de guardar la información de los productos que se prepara en la panadería de una sede, esta tabla, también tiene relación con la tabla de 'em\_gen\_empresa\_sede'.

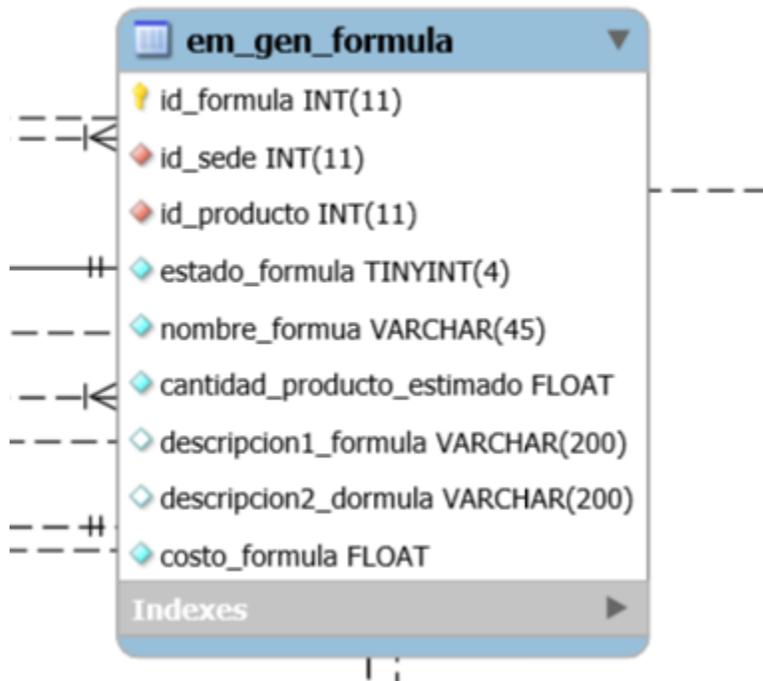


Figura 7.4.10

Autor: Autoría propia, herramienta usada: MySQL Workbench.

La tabla de 'em\_gen\_formula' se implementó con el objetivo de almacenar la información de las fórmulas que contienen los ingredientes que se necesitan para la elaboración de un producto. Se puede observar que la tabla, tiene dos claves foráneas, una para la relación con la tabla de 'em\_gen\_empresa\_sede' y otra con la tabla de 'em\_gen\_producto'.

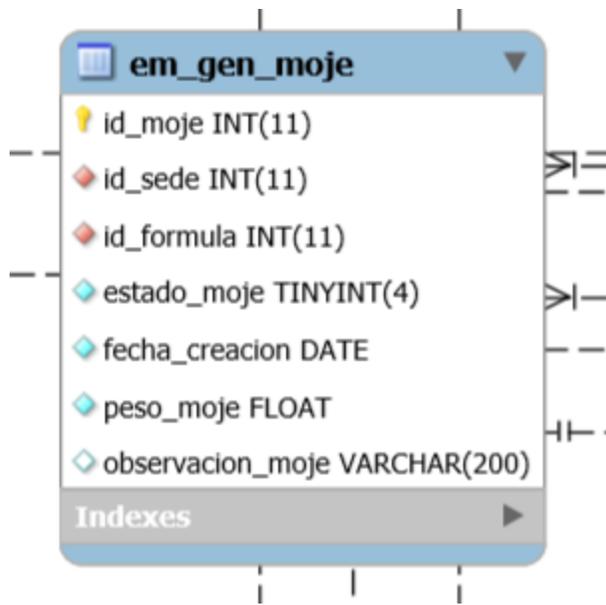


Figura 7.4.11

Autor: Autoría propia, herramienta usada: MySQL Workbench.

La tabla 'em\_gen\_moje' se realizó para almacenar la información de los mojes, esta tabla dos llaves foráneas, una de ellas es 'id\_formula' ya que el moje depende de una fórmula para su elaboración. La otra llave foránea es la relación con la tabla 'em\_gen\_empresa\_sede'.

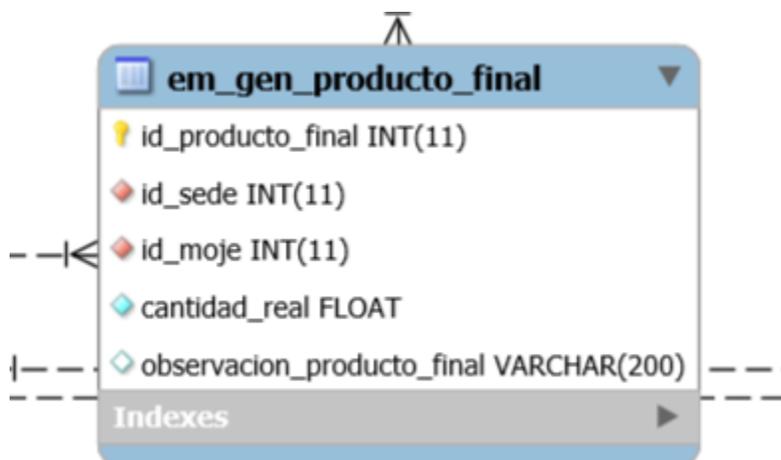


Figura 7.4.12

Autor: Autoría propia, herramienta usada: MySQL Workbench.

La tabla ‘em\_gen\_producto\_final’ se construyó con el fin de almacenar la información de un producto final, esta tabla guarda el resultado final de un moje, por eso la relación de la llave foránea con la tabla de ‘em\_gen\_moje’. La otra llave foránea es la relación con la tabla ‘em\_gen\_empresa\_sede’.

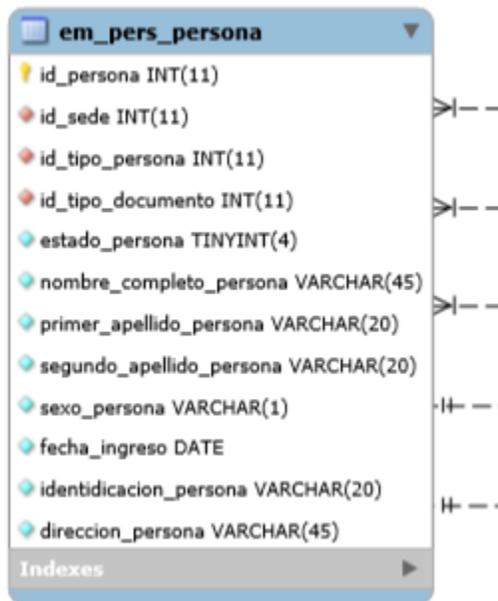


Figura 7.4.13

Autor: Autoría propia, herramienta usada: MySQL Workbench.

La tabla ‘em\_pers\_persona’ se realizó para almacenar la información mas importante de las personas que se ingresan a la aplicación. Esta tabla tiene una gran dependencia con otras tres tablas, hay tres claves foráneas donde la primera es la relación con la tabla de ‘em\_gen\_empresa\_sede’, la segunda es la relación con la tabla de ‘em\_pers\_tipo\_persona’ por medio de la clave foránea ‘id\_tipo\_persona’, la tercera llave foránea hace referencia a la relación con la tabla de ‘em\_pers\_tipo\_documento’.

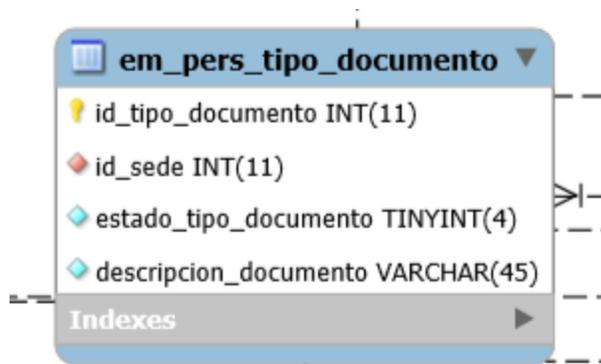


Figura 7.4.14

Autor: Autoría propia, herramienta usada: MySQL Workbench.

La tabla de ‘em\_pers\_tipo\_documento’ se implementó con el fin de almacenar la información referente a el tipo de documento que puede tener una persona. Esta tabla tiene relación con la tabla de de ‘em\_gen\_empresa\_sede’.

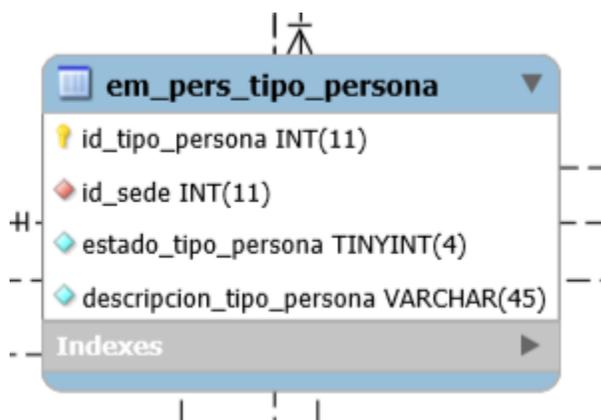


Figura 7.4.15

Autor: Autoría propia, herramienta usada: MySQL Workbench.

La tabla de ‘em\_pers\_tipo\_persona’ se realizó con el objetivo de guardar la información referente a los tipos de persona que puede tener una persona, ya sea natural o Jurídica. Esta tabla tiene relación con la tabla de ‘em\_gen\_empresa\_sede’.

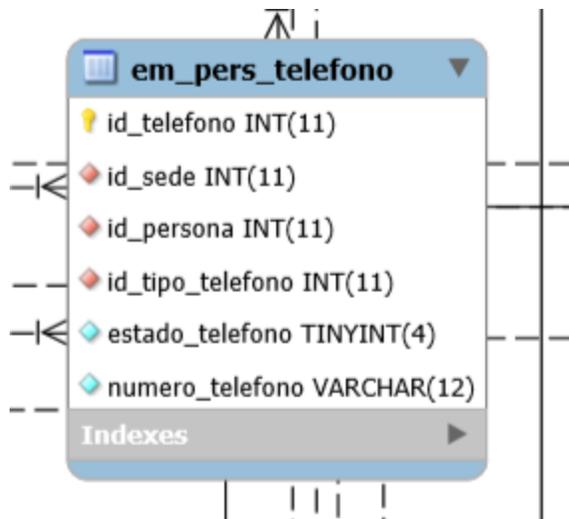


Figura 7.4.16

Autor: Autoría propia, herramienta usada: MySQL Workbench.

La tabla de 'em\_pers\_telefono' se creó para registrar la información referente a los teléfonos de una persona. La tabla tiene tres llaves foráneas, la primera de ellas es 'id\_sede' que hace referencia a la relación con la tabla de 'em\_gen\_empresa\_sede'. La segunda 'id\_persona' es el campo que hace relación con la tabla de 'em\_pers\_tipo\_persona'. La última llave foránea 'id\_tipo\_telefono' es la relación con la tabla de 'em\_pers\_tipo\_telefono', la cual se explicará a continuación.



Figura 7.4.17

Autor: Autoría propia, herramienta usada: MySQL Workbench.

La tabla ‘em\_pers\_tipo\_telefono’ se creó con el objetivo de guardar la información de los tipos de teléfonos que pueden asignarse a un teléfono de una persona. Un ejemplo de ellos es un teléfono fijo o un teléfono celular.

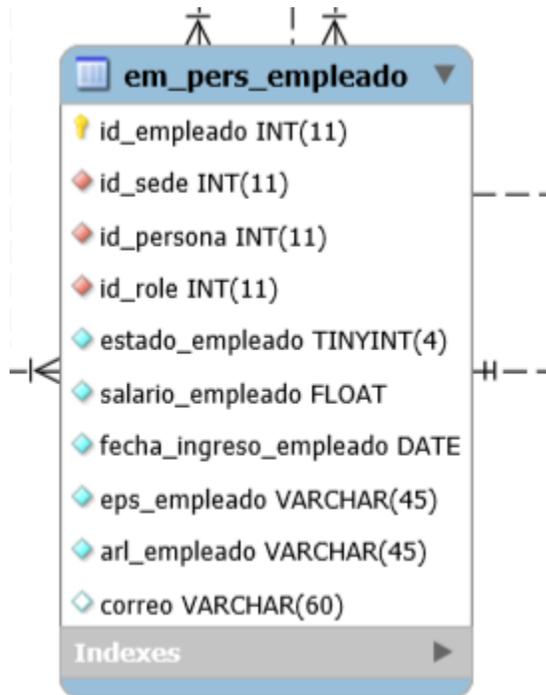


Figura 7.4.18

Autor: Autoría propia, herramienta usada: MySQL Workbench.

La tabla de ‘em\_pers\_empleado’ se generó con el fin de guardar toda la información relacionada con los empleados. Esta tabla contiene tres llaves foráneas que permiten saber cuál es la persona, de qué sede viene y cuál es el rol que tiene dicho empleado. Por ende, esta tabla tiene relación con las tablas de ‘em\_pers\_rol’, ‘em\_per\_persona’ y ‘em\_gen\_empresa\_sede’.

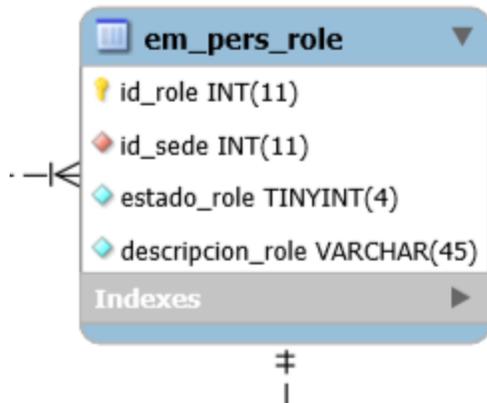


Figura 7.4.19

Autor: Autoría propia, herramienta usada: MySQL Workbench.

La tabla 'em\_pers\_role' se creó para guardar los roles que pueden tener los empleados de una sede. Para este proyecto, se decidió manejar tres tipos de roles:

- Super administrador
- Administrador
- Panadero.

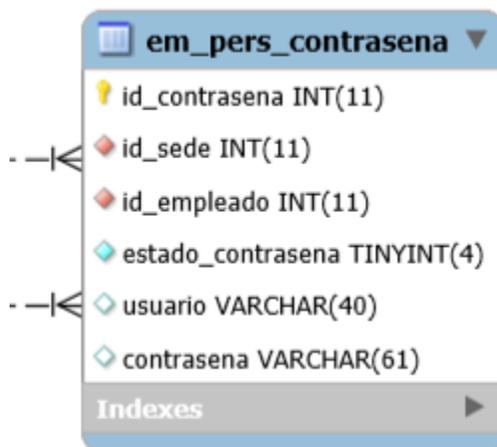


Figura 7.4.20

Autor: Autoría propia, herramienta usada: MySQL Workbench.

La tabla 'em\_pers\_contrasena' se implementó para guardar las credenciales de los empleados, es esta tabla se guardan atributos tan importantes como el usuario y a la contraseña con la cual un empleado pueden ingresar a la aplicación web. La tabla contiene dos relaciones, una con la tabal de 'em\_gen\_empresa\_sede' y otra con la tabla de 'em\_pers\_empleado'.

## 8. CONCLUSIONES

- Con la construcción de este proyecto se aprendieron diferentes tecnologías para el desarrollo de software, y lo más importante es que son tecnologías y herramientas que actualmente están usando las empresas de desarrollo de software, esto permitió que se ganara un poco de experiencia con el desarrollo de aplicaciones.
- El uso de Git permite realizar desarrollo de software de manera colaborativa, permitiendo trabajar diferentes personas en un mismo proyecto, de una manera controlada y organizada. Además, permite que el código fuente esté almacenado y centralizado en un solo lugar.
- Es de suma importancia aprender a escuchar al cliente, para poder determinar sus necesidades y así poder ofrecer una solución efectiva.
- Al realizar este proyecto se aprendió sobre conceptos de alimentos y los procesos de la panadería, desde una compra hasta la preparación de un producto final. Con esto, se termina de ratificar que la ingeniería de sistemas es una rama que permite aprender de cualquier tema ya sea conocido o desconocidos para el desarrollador.
- El software realizado puede contribuir mucho a disminuir el desperdicio de insumos y alimentos, ya que, con el control de inventarios, generación de fórmula y mojes, es posible tener control de los insumos que se están gastando diariamente en la panadería.
- El proyecto realizado puede contribuir a la toma de decisiones oportunas con base al valor del costo de una fórmula ya que esto permitirá saber si el precio del producto final que se está ofreciendo al cliente es rentable para la panadería. Además, con el porcentaje de pérdidas de

un producto final, permite determinar si el moje de un producto final generó ganancia o pérdida para la panadería.

- Con la construcción del proyecto se reforzó el trabajo en equipo, una habilidad blanda que fue fundamental para el cumplimiento del proyecto. Además, el trabajar de una manera ordenada y con objetivos claros, permitió cumplir con las expectativas del cliente.

## 9. RECOMENDACIONES

- Se recomienda utilizar frameworks para el desarrollo de las aplicaciones, ya que estos facilitan el desarrollo del software y permiten ahorrar tiempo, código y trabajo en la construcción del proyecto.
- Antes de empezar con el desarrollo del código de una aplicación, se recomienda primero definir las tecnologías con que se va a realizar el desarrollo. De igual manera, tener una definición clara de la arquitectura con que se construirá la aplicación, ya que esto permitirá trabajar sin retrocesos.
- Tener una arquitectura donde se separe el software por capas, es recomendable para que la aplicación tenga sus artefactos independientes y no genere dependencias.
- Implementar las herramientas más utilizadas en el mercado, es importante ya que permitirá aprender de ellas y adquirir a los desarrolladores más experiencia para cuando se enfrenten en el mundo laboral.
- Se recomienda tener un buen diseño de la base de datos desde el principio del proyecto, ya que esta es la estructura de la información de la aplicación, si está desde un principio se implementa de una manera correcta, evitará retrocesos cuando se esté desarrollando el software.

## 10. REFERENCIAS BIBLIOGRÁFICAS

Alvarez, M. A. (s.f de 01 de 2001). *desarrolloweb.com*. Obtenido de Qué es HTML:

<https://desarrolloweb.com/articulos/que-es-html.html>

Basalo, A. (12 de 05 de 2020). *desarrolloweb.com*. Obtenido de Angular CLI:

<https://desarrolloweb.com/articulos/angular-cli.html>

Benites, A. G. (s.f de s.f de s.f). *DevCode*. Obtenido de Frontend y Backend:

<https://devcode.la/blog/frontend-y-backend/>

Bootstrap team with the help of our contributors. (s.f de s.f de s.f). *getbootstrap*. Obtenido de

Build fast, responsive sites with Bootstrap: <https://getbootstrap.com/>

Caceres, M. (s.f de s.f de s.f). *DevCode*. Obtenido de ¿Qué es TypeScript?:

<https://devcode.la/blog/que-es-typescript/>

Garzas, J. (06 de 06 de 2014). *javiergarzas.com*. Obtenido de Simple y rápido. Entiende qué es

Maven en menos de 10 min.: <https://www.javiergarzas.com/2014/06/maven-en-10-min.html>

informaticamilenium. (s.f. de s.f. de s.f.). *informaticamilenium*. Recuperado el 04 de 2020, de

Definición de Software, Qué es, Significado y Concepto:

<https://www.informaticamilenium.com.mx/es/temas/que-es-software.html>

Muñoz, L. E. (2013). *Hágase Experto en Panadería y Repostería*. Pereira: Fondo Editorial de Risaralda.

Pahino, R. (31 de 03 de 2020). *CanpusMVP*. Obtenido de ¿Qué son Spring framework y Spring

Boot?: <https://www.campusmvp.es/recursos/post/que-son-spring-framework-y-spring-boot-tu-primer-programa-java-con-este-framework.aspx>

Rouse, M. (29 de 12 de 2017). *techtarget*. Obtenido de REST (REpresentational State Transfer):

<https://searcharchitecture.techtarget.com/definition/RESTful-API>

Zapata, J. F. (s.f de s.f de s.f). *DevCode*. Obtenido de ¿Qué es Responsive Web Design?:

<https://devcode.la/blog/que-es-responsive-web-design/>

