



UNIVERSIDAD DEL BÍO-BÍO

# Control y gestión de ventas e inventario para micro, pequeñas y medianas empresas

**Integrantes:** Evelyn Cariaga Sandoval  
Ailyn Cofré Sáez  
Nicolás Fuentes Salgado  
Felipe Muñoz Sánchez  
Agustín Pereira Castro  
Bruno Risso Osorio  
Felipe Sánchez Arnés

**Asignatura:** Ingeniería de Software  
**Grupo:** G03  
**Profesores:** Pedro Campos Soto  
Paola Legue Guerra  
**Fecha:** 25/10/2021

## Índice

1. Formulación.....	3
1.1. Problema.....	3
1.2. Propuesta.....	3
1.3. Alcance.....	4
2. Metodología de Trabajo.....	5
2.1. Objetivo General.....	5
2.2. Objetivos Específicos.....	5
3. Proceso de Desarrollo.....	6
3.1. Principales Elementos.....	6
3.2. Modelos de Proceso.....	8
3.3. Aplicabilidad del Modelo Principal.....	10
4. Requisitos.....	11
4.1. Principales Requisitos.....	11
4.2. Casos de Uso de Alto Nivel.....	11
5. Ambiente de Ingeniería de Software.....	14
5.1. Principales Elementos.....	14
6. Prueba de Concepto de Arquitectura.....	15
6.1. Principales Elementos.....	15
6.2. Evidencias y Resultado.....	15

## **1. FORMULACIÓN**

### **1.1. Problema**

En el contexto actual de crisis sanitaria, negocios tanto formales como informales, han tenido que digitalizarse para adaptarse y seguir operando. No obstante, este cambio no resulta sencillo para todos, en especial para los que comienzan a establecerse.

En el estudio ITD (Índice de Transformación Digital) 2020 en su tercera versión, se deja en evidencia que el aumento en la madurez digital de las organizaciones no fue homogéneo, sino más bien proporcional al nivel de madurez que tenía la empresa previo a la pandemia. Esto se debe a que no todas las personas manejan la tecnología a tal punto de depender de ella casi al 100% para trabajar. Es por eso que, uno de los mayores desafíos que tienen hoy las empresas, es el de potenciar las habilidades y competencias de su capital humano.

Así como hay empresas que lograron adoptar de buena forma este cambio, hay algunas que fracasaron, provocando el despido de muchos empleados. Algunas de estas personas en busca de una nueva fuente de ingresos, optan por empezar su pequeño emprendimiento y muchas veces no se tienen los conocimientos necesarios para poder llevar de buena manera el negocio. Y si bien existen diversos software que ofrecen una variedad de herramientas para organizar todo, resulta muy costoso contratar sus servicios teniendo en cuenta que lo ofrecido es mucho más de lo que necesitan y a la vez hace muy confuso su uso.

### **1.2. Propuesta**

Lo principal que desea todo negocio al comenzar es que sea rentable. Para ello, es necesario manejar de manera óptima la compra de productos e insumos, teniendo en cuenta diversos factores como las ventas por periodo y capacidad de almacenamiento.

Debido a lo anterior, queremos presentar EasyControl, una solución tanto para todas aquellas personas que desean levantar y mantener un negocio, como para aquellas con uno ya establecido, brindando una serie de funcionalidades capaces de solventar los principales problemas de gestión que puedan sufrir en el camino a ser una empresa estable.

Su facilidad de uso será su fortaleza, contando con una cómoda navegación que permitirá a quien lo usa, realizar el trabajo de forma rápida sin perderse ningún detalle de lo que está sucediendo.

Esto entregará una mayor seguridad en la administración, permitiéndoles a nuestros clientes enfocarse únicamente en sus productos. De esta forma, disminuirémos la cantidad de empresas que quiebran debido a la mala planificación de sus gastos y ahorros.

### **1.3. Alcance**

Esta será una aplicación web enfocada en gestionar el inventario y analizar las ventas. Para ello, el dueño o alguno de los empleados de la empresa deberá registrar todos los productos e insumos

con su información respectiva. Estos productos deberán estar identificados según una categoría, la cual deberá ser ingresada de forma manual para mayor flexibilidad.

En cuanto a la capacitación necesaria, éstas se harán mediante grabaciones de videos, donde se dará un recorrido por la aplicación para que todos estén al tanto de lo que se ofrece, y se explicará cada funcionalidad disponible.

De momento, en esta versión no se incluirán integraciones con otros sistemas, ya que planea enfocarse primeramente en lo básico que puede llegar a necesitar cualquier negocio: impacto del producto en el mercado; la rentabilidad y proyecciones financieras.

Este proyecto tiene una fecha de termino estimada para enero de 2022 y no se requerirá de ningún financiamiento, ya que se realizará con fines educativos.

## **2. METODOLOGÍA DE TRABAJO**

### **2.1. Objetivo General**

Entregar al usuario las herramientas necesarias para controlar su empresa en cuanto a la gestión de inventario y análisis financiero, manteniendo una simplicidad en la aplicación que permita ser usada por cualquiera con un conocimiento básico en informática, sin omitir características o información.

### **2.2. Objetivos Específicos**

- El usuario debe ser capaz de gestionar sus bodegas de forma independiente, de tal modo que su inventario no interfiera con el de otras bodegas.
- El inventario debe poder organizarse por categorías. Estas categorías pueden ser diferentes al de otras bodegas.
- La información financiera y de inventario debe mantenerse actualizada para todos los usuarios de la empresa registrada en cada cambio.
- No debe existir un mismo producto (mismas características) con diferente código de identificación.
- Se debe poder crear nuevos usuarios (para empleados) con distintos niveles de acceso a la información de la empresa.

Para cumplir con los objetivo, se deberá cumplir con las siguientes tareas:

- Crear la arquitectura, definir modelo y metodologías (fase de planificación y análisis de riesgo).
- Crear prototipo de página (fase de implementación).
- Construir base de datos con información ficticia (fase de implementación).
- Gestionar el acceso de usuarios y niveles de acceso (fase de implementación).
- Retrospectiva

Para monitorear el correcto funcionamiento, tenemos pensado pasar por una fase de testeo luego de cada objetivo.

### 3. PROCESO DE DESARROLLO

#### 3.1. Principales Elementos

Actividades	Tareas	Roles
1. Búsqueda y análisis de características más viables.	<ul style="list-style-type: none"><li>• Buscar información referente a las principales características que necesitan las empresas para gestionar su negocio.</li><li>• Elegir las principales características en base a la información recopilada anteriormente.</li></ul>	<ul style="list-style-type: none"><li>• Todos los integrantes del equipo deben realizar esta actividad.</li></ul>
2. Establecimiento de incrementos.	<ul style="list-style-type: none"><li>• Establecer los objetivos de cada incremento.</li><li>• Lograr que los avances de cada incremento se logren entregar en los tiempos estipulados.</li></ul>	<ul style="list-style-type: none"><li>• Todos los integrantes del equipo deben realizar esta actividad.</li></ul>
3. Diseño del sistema.	<ul style="list-style-type: none"><li>• Elegir las funcionalidades que tendrá la aplicación.</li><li>• Identificar las entidades que estarán involucradas.</li><li>• Realizar modelo del sistema.</li></ul>	<ul style="list-style-type: none"><li>• El diseñador será quien estará encargado de realizar esta actividad, pero también puede ser apoyado por otros integrantes del equipo.</li></ul>
4. Codificación del sistema.	<ul style="list-style-type: none"><li>• Preparación del ambiente de trabajo.</li><li>• Crear comunicación con servidor.</li></ul>	<ul style="list-style-type: none"><li>• Los encargados serán los programadores, quienes estarán supervisados por el</li></ul>

	<ul style="list-style-type: none"> <li>• Crear una base de datos.</li> <li>• Desarrollo de Front-end</li> <li>• Desarrollo de Back-end</li> </ul>	diseñador de software.
5. Pruebas en las distintas capas.	<ul style="list-style-type: none"> <li>• Realizar pruebas necesarias.</li> <li>• Verificar que se cumplan los requisitos.</li> <li>• Retroceder en alguna iteración de ser necesario.</li> </ul>	<ul style="list-style-type: none"> <li>• El Tester estará encargado de realizar las pruebas y le notificará al resto del equipo en caso de presentarse alguna falla en la implementación.</li> </ul>

Para la realización del proyecto, se establecieron tres etapas por separado y que deben ser entregadas sin falta para poder cumplir con los tiempos estipulados.

- **Entrega N° 1 (Formulación de Proyecto)**

Esta primera entrega consta de un informe el cual debe incluir los requisitos ya establecidos para el proyecto, metodología que se va a emplear, proceso de desarrollo en donde se presenten sus principales elementos, tales como actividades y tareas a desarrollar, los roles que tomarán los diferentes integrantes del equipo, y la arquitectura del sistema. Además de esto se debe presentar una prueba de concepto en base a la arquitectura propuesta, para de este modo poder familiarizarse con las tecnologías que se van a utilizar y dejar preparado el ambiente de trabajo.

- **Entrega N° 2 (Primer Incremento de Software)**

Para esta entrega se debe tener un prototipo funcional del proyecto que integre los principales requisitos y casos de uso ya establecidos, junto con la documentación que incluya el funcionamiento general del sistema.

- **Entrega N° 3 (Segundo Incremento de Software)**

Se debe tener un prototipo más avanzado y mejorado, junto con los cambios realizados en el caso de ser necesario.

### **3.2. Modelos de Proceso**

Usaremos el modelo en espiral, que describe el ciclo de vida de un software por medio de espirales, los cuales se repiten hasta que se entrega el producto terminado. Se caracteriza por tener la capacidad de minimizar los riesgos en el desarrollo de un software, evitando posibles aumentos en los costes o algún retraso en el lanzamiento del software. Además, el modelo en espiral es genérico, lo cual le permite combinarse con otros métodos de desarrollo ágil.

La elección se debe a que se caracteriza por ser un modelo flexible que permite realizar cambios durante el desarrollo del proyecto, sin que este pierda estabilidad o tenga algún retraso inesperado. Además, al ser un proyecto de un tamaño considerable, este modelo se ajusta perfectamente a él debido a su naturaleza iterativa.

Este tipo de desarrollo se divide en tres iteraciones y cuatro fases:

#### **Iteración 1: Plan de desarrollo**

El objetivo de esta iteración es organizar el equipo de desarrollo; determinar objetivos, requerimientos y el entorno de desarrollo, además de metodologías a utilizar para desarrollar el software. El resultado de esta iteración es el documento presente y corresponde a la primera entrega de la asignatura.

- Determinar objetivos
  - Se piensa en los objetivos a realizar y a qué pasos del desarrollo se van a vincular. Al mismo tiempo, se discuten posibles alternativas y se especifican las condiciones marco, como los costes, tiempo invertido, sistemas operativos a utilizar, entornos y lenguajes de programación.
- Identificar y resolver riesgos
  - Se establecen los requerimientos del proyecto además del plan y los métodos de desarrollo. Los riesgos no se resolverán debido a que a la fecha de la creación de este documento, no ha sido vista en el curso.
- Desarrollo y testeo
  - Se realizó una prueba de concepto en la que se probó el funcionamiento de los componentes que usaremos durante el desarrollo.



### **Iteración 2: Prototipo inicial**

El objetivo de la segunda iteración es crear un prototipo de la página web sin conexión a la base de datos.

### **Iteración 3: Prototipo avanzado**

El objetivo principal de la tercera iteración es crear la base de datos utilizada para la página web y realizar la conexión entre estos dos elementos. Esto representa la segunda entrega de la asignatura.

### **Iteración 4: Prototipo funcional**

Para la cuarta iteración, mejoraremos el prototipo conectado a la base de datos y añadiremos funcionalidades a la aplicación. El resultado de esta entrega corresponde a la última de la asignatura.

En cuanto a las fases del modelo, tenemos las siguientes:

#### **Fase 1: Determinar objetivos**

Se piensa en los objetivos a realizar y a qué pasos del desarrollo se van a vincular. Al mismo tiempo, se discuten posibles alternativas y se especifican las condiciones marco como los costes, tiempo invertido, sistemas operativos a utilizar, entornos y lenguajes de programación.

#### **Fase 2: Plan de requerimientos y concepto de operación**

Se establecen los requerimientos del proyecto (definidos en detalle más adelante en este documento) además del plan y los métodos de desarrollo.

#### **Fase 3: Desarrollo y prueba**

Los prototipos se amplían y se añaden funcionalidades. El código real es escrito, probado y migrado a un entorno de prueba varias veces hasta que el software pueda ser implementado en un entorno productivo.

#### **Fase 4: Planificación del siguiente ciclo**

El siguiente ciclo se planifica al final de cada etapa. Si se producen errores, se buscan soluciones, y si una alternativa es una mejor solución, se prefiere en el siguiente ciclo.

### 3.3. Aplicabilidad del Modelo Principal

Modelo	Ventajas	Desventajas
1. Cascada	<ul style="list-style-type: none"><li>• Fácil planificación.</li><li>• Se pueden estimar los costes antes de empezar el proyecto.</li><li>• Permite tener mayor claridad del cómo será la estructura del proyecto.</li></ul>	<ul style="list-style-type: none"><li>• No permite realizar ajustes durante el desarrollo en caso de ser necesario.</li><li>• Las fallas se logran detectar casi al finalizar el proyecto.</li></ul>
2. Incremental	<ul style="list-style-type: none"><li>• Reduce el tiempo de desarrollo al implementar la funcionalidad parcial.</li><li>• Resulta más sencillo realizar cambios.</li><li>• Se centra en la entrega de un producto operativo en cada incremento.</li></ul>	<ul style="list-style-type: none"><li>• Se requiere de una experiencia previa para definir los requerimientos.</li><li>• Si se cuenta con solo un grupo para construirlo, se corre el riesgo de que el desarrollo se prolongue demasiado.</li></ul>
3. Proceso unificado	<ul style="list-style-type: none"><li>• Permite deducir riesgos.</li><li>• Permite que sea más fácil realizar cambios a futuro.</li></ul>	<ul style="list-style-type: none"><li>• Se debe tener una previsión de lo que ocurrirá. Esto puede generar trabajo adicional.</li></ul>

## 4. REQUISITOS

### 4.1. Principales Requisitos

- El sistema debe registrar los datos del usuario.
- Permitir el inicio de sesión al sistema con email y contraseña.
- El usuario administrador puede editar los usuarios de su empresa.
- Permitir modificaciones en la información de los usuarios.
- Permitir al usuario gestionar su cuenta.
- Permitir la creación de varias bodegas y visualizar su inventario individualmente.
- Permitir modificación de un producto asociado a una bodega.
- Permitir la exportación del inventario a un archivo Excel.
- La aplicación mostrará tutorial paso a paso con el fin de capacitar al usuario hacia las nuevas tecnologías.
- Se requiere de un registro mínimo de un mes de ventas para poder generar proyecciones.
- El producto debe contar con su información básica (id, nombre, categoría, valor y medida) para poder registrarse.
- La bodega debe contar con su información básica (nombre, descripción, dirección y capacidad) para poder registrarse.
- Se debe registrar cualquier actividad realizada por cualquier usuario.

### 4.2. Casos de Uso de Alto Nivel

<b>Caso de uso</b>	Registrar usuarios
<b>Actores</b>	Usuario administrador
<b>Objetivo</b>	Permitir al administrador crear nuevos usuarios en su empresa
<b>Descripción</b>	El administrador ingresa a la sección para crear un nuevo usuario. Verifica su rol ingresando su contraseña. Llena un formulario de registro con los datos del empleado a añadir. Registra el formulario y se envían las credenciales de acceso al correo del empleado ingresado.
<b>Caso de uso</b>	Iniciar sesión
<b>Actores</b>	Cualquier usuario registrado
<b>Objetivo</b>	Ingresar a la aplicación web
<b>Descripción</b>	El usuario ingresa a la sección de inicio de sesión del sitio web. Ingresar su

	email y contraseña para autenticarse. Si es correcto, redirecciona a la página de bienvenida.
--	-----------------------------------------------------------------------------------------------

<b>Caso de uso</b>	Cambia contraseña
<b>Actores</b>	Cualquier usuario registrado
<b>Objetivo</b>	Cambiar la contraseña de usuario
<b>Descripción</b>	Ingresar al área de Mi Cuenta. Ubicar la sección de cambiar contraseña, en información personal. Ingresar la contraseña actual y la nueva con su respectiva confirmación. Aceptar los cambios.

<b>Caso de uso</b>	Administrar privilegios de usuarios
<b>Actores</b>	Usuario administrador
<b>Objetivo</b>	Modificar los privilegios de los usuarios de su empresa
<b>Descripción</b>	Ubicar la lista de empleados de la empresa. Seleccionar el usuario a modificar. Cambiar sus privilegios. Confirmar cambios.

<b>Caso de uso</b>	Eliminar usuarios
<b>Actores</b>	Usuario administrador
<b>Objetivo</b>	Eliminar un usuario de la empresa
<b>Descripción</b>	Ubicar la lista de empleados de la empresa. Eliminar usuario. Confirmar eliminación.

<b>Caso de uso</b>	Gestionar cuenta de usuario
<b>Actores</b>	Cualquier usuario registrado
<b>Objetivo</b>	Modificar cualquier información personal
<b>Descripción</b>	Ir a Mi Cuenta. Editar los campos (ya precargados con la información actual) que se deseen modificar. Confirmar cambios realizados.

<b>Caso de uso</b>	Añadir, actualizar o eliminar bodega
<b>Actores</b>	Usuarios con el privilegio para añadir, actualizar y eliminar una bodega
<b>Objetivo</b>	Añadir, actualizar o eliminar una bodega
<b>Descripción</b>	Ingresa a la lista de bodegas existentes. Añadir, actualizar o eliminar una bodega de la lista. Confirmar.

<b>Caso de uso</b>	Añadir o actualizar producto
<b>Actores</b>	Usuarios con el privilegio para añadir y actualizar producto
<b>Objetivo</b>	Añadir o actualizar un producto
<b>Descripción</b>	Ingresa a la lista de productos de la empresa. Añadir o actualizar un producto de la lista. Confirmar.

<b>Caso de uso</b>	Visualizar tutoriales
<b>Actores</b>	Cualquier usuario registrado
<b>Objetivo</b>	Capacitarse en la aplicación mediante la visualización de videotutoriales
<b>Descripción</b>	Ubicar cualquier funcionalidad que ofrezca mayor información (icono). Mostrar dicha información. Ver el video.

<b>Caso de uso</b>	Acceso a resumen financiero
<b>Actores</b>	Usuarios con el privilegio para ver información financiera
<b>Objetivo</b>	Visualizar un resumen financiero general
<b>Descripción</b>	En el panel de la aplicación, ingresar a finanzas.

<b>Caso de uso</b>	Exportar
<b>Actores</b>	Cualquier usuario registrado
<b>Objetivo</b>	Exportar información a un archivo
<b>Descripción</b>	Ubicar cualquier información con la opción de exportar. Seleccionar el formato destino (PDF o Excel).

## **5. AMBIENTE DE INGENIERÍA DE SOFTWARE**

### **5.1. Principales Elementos**

Para el Front-end, usaremos el framework Bootstrap 5, usaremos JavaScript en conjunto con CSS para lograr una estética bien trabajada.

En el lado del Back-end, se usará Laravel. Este es un framework de PHP de código abierto para desarrollar aplicaciones web de forma personalizada de gran calidad.

Con el apoyo de Git, sincronizamos las estaciones de trabajo del equipo y pasaremos las pruebas al servidor.

En el ambiente de trabajo, usaremos como apoyo Trello para organizar las actividades, Lucid para realizar los diagramas y Postman para testear la funcionalidad.

La documentación de los procesos será en formato HIPO, ya que es más fácil representar y transmitir información.

## 6. PRUEBA DE CONCEPTO DE ARQUITECTURA

### 6.1. Principales Elementos

La prueba de concepto consiste en un CRUD realizado en la tabla 'bodegas'. Las funciones del CRUD fueron repartidas por la página usando rutas URL donde se realizan dichas funciones por separado, manteniendo la navegabilidad.

### 6.2. Evidencias y Resultado

```
s > views > bodega > index.blade.php
<a href="{{ url('/') }}"> Atras</a>
<br>
<a href="{{ url('bodega/create') }}"> Registrar nueva bodega</a>
<br><br>

Lista de las primeras 50 bodegas de la bdd
<br>

<table class="table table-light">
  <thead class="thead-light">
    <tr>
      <th>#</th>
      <th>Nombre</th>
      <th>Ubicacion</th>
      <th>Acciones</th>
    </tr>
  </thead>

  <tbody>
    @foreach( $bodegas as $bodega )
      <tr>
        <td>{{ $bodega->id }}</td>
        <td>{{ $bodega->nombre }}</td>
        <td>{{ $bodega->ubicacion }}</td>
        <td>
          <a href="{{ url('/bodega/' . $bodega->id . '/edit') }}">
            Editar
          </a>
          <td>
            <form action="{{ url('/bodega/' . $bodega->id ) }}" method="post">
              @csrf
              {{ method_field('DELETE')}}
              <input
                type="submit"
                onclick="return confirm('Quieres borrar esta bodega?')"
                value="Borrar"
              >
            </form>
          </td>
        </tr>
      @endforeach
    </tbody>
  </table>
```

Index de la bodega, maneja la lectura y borrado para el CRUD

```

ces > views > bodega > create.blade.php
<a href="{{ url('/bodega') }}"> Atras</a>
<br><br>

Formulario creacion de bodega
<br>

<form action="{{ url('/bodega') }}" method="post" enctype="multipart/form-data">
@csrf

@include('bodega.form')

</form>

```

Create de la bodega, maneja la creación para el CRUD

```

ces > views > bodega > edit.blade.php
<a href="{{ url('/bodega') }}"> Atras</a>
<br><br>

Formulario edicion de bodega
<br>

<form action="{{ url('/bodega/' . $bodega->id) }}" method="post">
@csrf
{{method_field('PATCH')}}

@include('bodega.form')

</form>

```

Edit de la bodega, maneja la edición para el CRUD



```

s > views > bodega > form.blade.php

<label for="Nombre"> Nombre</label>
<input type="text" name="Nombre" value="{{ isset($bodega) ? $bodega->nombre : '' }}" id="Nombre">
<br>
<label for="Ubicacion"> Ubicacion</label>
<input type="text" name="Ubicacion" value="{{ isset($bodega) ? $bodega->ubicacion : '' }}" id="Ubicacion">
<br>
<input type="submit" value="Guardar bodega">
<br>

```

Formulario necesario para las vistas de Create y Edit

```

s > views > welcome.blade.php
<DOCTYPE html>
<html lang="{{ str_replace('_', '-', app()->getLocale()) }}">
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">

<title>Easy Control</title>

<!-- Fonts -->
<link href="https://fonts.googleapis.com/css2?family=Nunito:wght@400;600;700&display=swap" rel="stylesheet">

<!-- Styles -->
<style>...
</style>

<style>...
</style>
</head>
<body class="antialiased">
<div class="relative flex items-top justify-center min-h-screen bg-gray-100 dark:bg-gray-900 sm:items-center py-4 sm:pt-0">
    @if (Route::has('login'))
        <div class="hidden fixed top-0 right-0 px-6 py-4 sm:block">
            @auth
                <a href="{{ url('/home') }}" class="text-sm text-gray-700 dark:text-gray-500 underline">Home</a>
            @else
                <a href="{{ route('login') }}" class="text-sm text-gray-700 dark:text-gray-500 underline">Log in</a>

                @if (Route::has('register'))
                    <a href="{{ route('register') }}" class="ml-4 text-sm text-gray-700 dark:text-gray-500 underline">Register</a>
                @endif
            @endauth
        </div>
    @endif

    <div class="max-w-6xl mx-auto sm:px-6 lg:px-8">
        <div class="flex justify-center pt-8 sm:justify-start sm:pt-0">
            <svg viewBox="0 0 651 192" fill="none" xmlns="http://www.w3.org/2000/svg" class="h-16 w-auto text-gray-700 sm:h-20">
                <g clip-path="url(#clip0)" fill="#EF3B2D">
                    <path d="M248.032 44.676h-16.466v100.23h47.394v-14.748h-30.928V44.676z" data-bbox="248.032 44.676 338.77 100.23">
                </g>
            </svg>
        </div>

        <div class="mt-8 bg-white dark:bg-gray-800 overflow-hidden shadow sm:rounded-lg">
            <div class="grid grid-cols-1 md:grid-cols-2">
                <div class="p-6">

```

Vista principal del proyecto, página de bienvenida

```

class CreateBodegasTable extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::create('bodegas', function (Blueprint $table) {
            $table->id();

            $table->string('nombre');
            $table->string('ubicacion');

            $table->timestamps();
        });
    }

    /**
     * Reverse the migrations.
     *
     * @return void
     */
    public function down()
    {
        Schema::dropIfExists('bodegas');
    }
}

```

Creación de tabla 'bodegas' mediante migración