Imagen que contiene Texto

Descripción generada automáticamente

Jaime Medina

Eva Gómez

Jorge Benavente

David Jaraba

**TIENDA NEST**

Índice

[Índice 2](#_Toc160697132)

[Introducción 3](#_Toc160697133)

[Funcionalidades principales 3](#_Toc160697134)

[Gestión de Productos 3](#_Toc160697135)

[Gestión de Proveedores 4](#_Toc160697136)

[Gestión de Clientes 4](#_Toc160697137)

[Gestión de Empleados 4](#_Toc160697138)

[Gestión de Categorías 5](#_Toc160697139)

[Gestión de Pedidos 5](#_Toc160697140)

[Gestión de Usuarios 5](#_Toc160697141)

[Precios 6](#_Toc160697142)

[Especificaciones técnicas para las bases de datos 6](#_Toc160697143)

[PostgreSQL 6](#_Toc160697144)

[MongoDB 6](#_Toc160697145)

[Propuesta recomendada para la aplicación 7](#_Toc160697146)

[Ventajas de la propuesta recomendada 8](#_Toc160697147)

[Consideraciones de Seguridad y Confiabilidad 8](#_Toc160697148)

[Servicios recomendados 9](#_Toc160697149)

[Elastic Load Balancing 9](#_Toc160697150)

[Servicio de Almacenamiento Simple (S3) 9](#_Toc160697151)

[Estimación de costes 10](#_Toc160697152)

[Costes de Producción 10](#_Toc160697153)

[Elección de tecnologías 12](#_Toc160697154)

[PHP con Laravel 12](#_Toc160697155)

[PostgreSQL 12](#_Toc160697156)

[MongoDB 13](#_Toc160697157)

[DIAGRAMA ENTIDAD RELACIÓN 15](#_Toc160697158)

Introducción

En el dinámico mundo de la tecnología, la gestión eficiente es crucial para el éxito de cualquier negocio, especialmente en el sector de la informática. La capacidad de manejar de manera efectiva inventarios, proveedores, clientes y empleados es esencial para garantizar operaciones fluidas y satisfacer las demandas del mercado.

Con este propósito, hemos desarrollado una aplicación innovadora utilizando Laravel, un framework de PHP conocido por su elegancia y potencia en el desarrollo web. Esta aplicación está meticulosamente diseñada para abordar las necesidades de gestión de una tienda de informática, ofreciendo una solución óptima que simplifica la administración de productos, proveedores, clientes y empleados.

En esta presentación, les mostraré las características clave de nuestra aplicación y cómo Laravel nos ha permitido construir una plataforma confiable y flexible para optimizar las operaciones comerciales. Desde la gestión de inventarios hasta la administración de clientes y personal, nuestra aplicación ofrece una experiencia completa que impulsa el crecimiento y la eficiencia de cualquier tienda de informática.

Funcionalidades principales

Gestión de Productos

Para la gestión de productos en nuestra aplicación, utilizamos una entidad **Product** que nos permite almacenar información detallada sobre cada artículo en nuestro inventario. Aquí está cómo se gestionan los productos y algunas de las operaciones disponibles:

Descripción de cómo se gestionan los productos y sus categorías asociadas:

* Cada producto está representado por una instancia de la clase **Product**, que contiene atributos como nombre, peso, precio, imagen, stock y descripción.
* Para asociar un producto con una categoría, utilizamos una relación de muchos a uno (**ManyToOne**) con la entidad **Category**. Esto nos permite organizar nuestros productos en diferentes categorías para una mejor organización y navegación.
* Del mismo modo, asociamos cada producto con un proveedor utilizando una relación de muchos a uno (**ManyToOne**) con la entidad **Supplier**. Esto nos ayuda a rastrear qué proveedor suministra cada artículo en nuestro inventario.

Algunas de las operaciones que podemos realizar en nuestro programa por lo tanto serían crear, actualizar o borrar un producto, verificando anteriormente que las entidades relacionadas, como son **Supplier** y **Category,** existan en la base de datos. Además, podemos asignarle una imagen al producto que no sea la que viene por defecto, si ya tenía una imagen anteriormente que no fuese la de por defecto, esta se eliminaría de la carpeta donde están almacenadas y se sustituiría por la nueva, para ello tendremos una gestión de almacenamiento.

Gestión de Proveedores

En esta entidad tendremos información relevante sobre cada proveedor, como su nombre, contacto, dirección, fecha de contratación, categoría y productos asociados.

Tendrá un id generado con UUID ya que nuestro interés en esta entidad viene en los productos asociados, con los que tendrá una relación uno a muchos, esto significa que cada proveedor puede suministrar múltiples productos de la tienda.

Esta relación permite un seguimiento efectivo de los proveedores y los productos que suministran, lo que garantiza una gestión eficiente de los recursos.

Gestión de Clientes

En nuestra aplicación de gestión de clientes, hemos implementado un sistema de autenticación basado en Laravel con middlewares y la función Auth(). Esto significa que, para acceder a cualquier funcionalidad de la aplicación, incluida la lista de productos, los clientes deben autenticarse correctamente utilizando las herramientas de autenticación proporcionadas por Laravel.

Una vez que un cliente inicia sesión correctamente utilizando sus credenciales, el sistema verifica su rol de usuario para asegurarse de que tengan los permisos adecuados para acceder a la lista de productos. Si el rol del usuario es 'client', se les permite acceder a la ruta correspondiente que muestra la lista de productos disponibles.

La utilización del sistema de autenticación y autorización de Laravel garantiza que solo los usuarios autorizados, en este caso los clientes con el rol adecuado puedan acceder a la funcionalidad de ver los productos disponibles. Esto mejora significativamente la seguridad y la protección de los datos de la aplicación, al tiempo que proporciona una experiencia de usuario segura y controlada.

Gestión de Empleados

En nuestra aplicación, todos los empleados tienen acceso completo a todas las funcionalidades, lo que les permite realizar operaciones CRUD en productos, clientes y otros recursos de la tienda de informática. Al igual que los clientes, los empleados también deben autenticarse en la aplicación.

Utilizando middlewares y la función Auth() de Laravel, verificamos en cada solicitud si el empleado está autenticado y tiene el rol adecuado para acceder a la funcionalidad solicitada. Si el empleado no está autorizado para realizar una determinada acción, se deniega el acceso y se devuelve al inicio de sesión.

Esta implementación garantiza que solo los empleados autenticados y autorizados puedan acceder y operar en todas las funcionalidades de la aplicación, lo que mejora significativamente la seguridad y protección de los datos de la tienda de informática.

Gestión de Categorías

Cada categoría está representada por una instancia de la clase **Category**, que contiene atributos como nombre, fecha de creación, fecha de actualización y estado de activación. El nombre de la categoría se utiliza para identificarla de manera única y es obligatorio. El estado de activación indica si la categoría está activa o no. Por defecto, las categorías se crean como activas, pero este estado puede modificarse según sea necesario.

Cada categoría puede estar asociada con múltiples productos a través de una relación de uno a muchos. Esto nos permite asignar productos a categorías específicas para una mejor organización.

Asimismo, una categoría puede estar asociada con múltiples proveedores a través de otra relación de uno a muchos. Esto puede ser útil para rastrear los proveedores que suministran productos dentro de una categoría particular.

Gestión de Pedidos

Cada pedido contiene atributos como el id, el id del usuario que realizó el pedido, el cliente asociado, las líneas de pedido, el total de elementos, el total del pedido, la fecha de creación, la fecha de actualización y el estado de eliminación.

El cliente asociado al pedido contiene información como el nombre, el correo electrónico, el teléfono y la dirección de envío del cliente.

La dirección contiene atributos como la calle, el número, la ciudad, la provincia y el código postal.

Cada línea de pedido contiene detalles como la cantidad, el identificador único del producto, el precio unitario del producto y el total de esa línea de pedido.

El total de elementos y el total del pedido se calculan en base a las líneas de pedido incluidas en el pedido.

Cada pedido está asociado a un cliente a través de la entidad **Client**, que proporciona información detallada sobre el cliente que realizó el pedido. Además, cada pedido puede contener múltiples líneas de pedido.

Gestión de Usuarios

Además, en esta nueva versión de la tienda hemos incluido un carrito, en él puedes ver los productos añadidos y sus datos. Damos la opción de comprar, para ello debes rellenar datos como la dirección de envío y el número de tarjeta. Finalmente, al comprar, podrás ver los datos de tus compras en la ventana Home. En la que además cada usuario podrá ver sus propios datos y editarlos si es necesario.

Precios

Especificaciones técnicas para las bases de datos

PostgreSQL

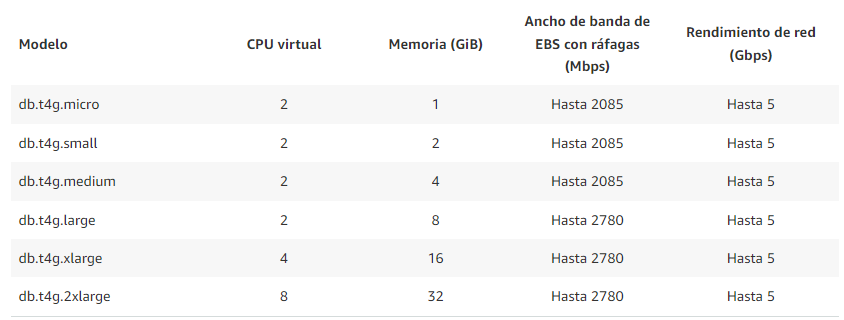
Para alojar nuestras bases de datos, hemos decidido utilizar las instancias RDS para PostgreSQL disponibles en AWS. En particular, para el entorno de producción, recomendamos comenzar con la instancia **db.t4g.medium**. Esta instancia proporciona una base de datos con las siguientes características:

* CPU virtual: 2
* Memoria RAM: 4GB
* 30 GB de almacenamiento SSD

En caso de que nuestras necesidades de base de datos aumenten, tenemos la flexibilidad de mejorar la instancia fácilmente. Al igual que con las instancias EC2, podemos optar por una instancia de mayor capacidad en momentos específicos de manera simple y directa.

También existen otras opciones dentro de AWS como T3, T2, M7g… pero hemos elegido la T4g ya que es la que más nos conviene para nuestra aplicación por precio, rendimiento y escalabilidad.

Dentro de T4g existen distintas opciones a las que podemos optar en caso de aumento de demanda.



Si deseas consultar las distintas opciones disponibles respecto a la elección del sistema de alojamiento de base de datos acceda al siguiente enlace: [AMAZON](https://aws.amazon.com/es/rds/instance-types/) RDS

MongoDB

Optaremos por utilizar las instancias de Amazon DocumentDB, que son compatibles con MongoDB. Estas instancias operan bajo un modelo de pago por uso, lo que significa que no hay costos iniciales; en su lugar, pagamos según el almacenamiento utilizado, las operaciones de entrada/salida y el tráfico de datos. En esta situación, al igual que con RDS, recomendaremos la instancia **db.t4g.medium**, que ofrece las mismas características:

* CPU virtual: 2
* Memoria RAM: 4GB

Al igual que las otras instancias, está también es igual de escalable con estas opciones superiores

Interfaz de usuario gráfica

Descripción generada automáticamente

Si deseas consultar las distintas opciones disponibles respecto a la elección del sistema de alojamiento de base de datos acceda al siguiente enlace: [AMAZON DocumentDB](https://aws.amazon.com/es/documentdb/pricing/)

Propuesta recomendada para la aplicación

La idea es utilizar los servidores de AWS para alojar nuestra aplicación en producción. Optamos por AWS porque ofrece una variedad de servidores y facilita la escalabilidad horizontal, lo que significa que podemos aumentar la capacidad de nuestra aplicación de manera flexible para adaptarnos a cambios en la demanda. Además, confiamos en la fiabilidad y el rendimiento de la infraestructura de AWS.

Para comenzar, para aprovechar al máximo la infraestructura de AWS, planeamos implementar nuestra aplicación en una instancia EC2. Estas instancias proporcionan máquinas virtuales donde podemos ejecutar nuestras aplicaciones. Para nuestro caso específico, hemos elegido una instancia **t4g.medium**, que tiene las siguientes características:

* CPU virtual: 2
* Memoria RAM: 4GB
* Ancho de banda: hasta 5Gb por segundo

En situaciones en las que la demanda de nuestra aplicación aumente, tenemos la flexibilidad de mejorar la instancia actual a una de mayor capacidad o aumentar el número de instancias que estamos utilizando. A continuación, mencionamos las instancias a las que podríamos escalar partiendo de instancia **t4g.medium:**

Captura de pantalla de computadora

Descripción generada automáticamente

Si desea consultar dicha información diríjase al siguiente enlace: [AMAZON EC2](https://aws.amazon.com/es/ec2/instance-types/t4/)

Ventajas de la propuesta recomendada

La instancia EC2 t4g.medium se destaca como la mejor opción para nuestra aplicación en este momento por varias razones. Ofrece un equilibrio óptimo entre rendimiento y costo, lo que nos permite maximizar la eficiencia de nuestros recursos.

Además, su capacidad para manejar la carga actual de la aplicación garantiza un funcionamiento fluido y sin interrupciones. La flexibilidad inherente de esta instancia nos permite escalar nuestros recursos según sea necesario, lo que nos brinda la tranquilidad de adaptarnos fácilmente a cambios en la demanda sin comprometer el rendimiento.

Consideraciones de Seguridad y Confiabilidad

La elección de AWS como nuestra plataforma de alojamiento no solo se basa en consideraciones de rendimiento y escalabilidad, sino también en su destacada reputación en seguridad y confiabilidad.

AWS ofrece una amplia gama de herramientas y servicios diseñados para proteger nuestros datos y garantizar la integridad de nuestra aplicación. Esto incluye características como cortafuegos, controles de acceso, monitoreo en tiempo real y protección contra ataques DDoS. Además, la infraestructura de AWS está respaldada por una sólida red de centros de datos y una infraestructura global, lo que garantiza una alta disponibilidad y redundancia para nuestra aplicación.

Servicios recomendados

Elastic Load Balancing

Si nuestra aplicación requiere escalar, es altamente aconsejable utilizar el servicio de Elastic Load Balancing. Elastic Load Balancing (ELB) es un servicio de AWS diseñado para distribuir automáticamente el tráfico entrante de las aplicaciones en múltiples instancias de Amazon EC2, contenedores de Amazon ECS o direcciones IP de clientes en varias zonas de disponibilidad.

Tabla

Descripción generada automáticamente

Estas son algunas de sus características:

* **Alta disponibilidad**: ELB garantiza la fiabilidad al redirigir el tráfico solo a instancias saludables.
* **Escalabilidad automática**: Ajusta el número de instancias EC2 según la carga de la aplicación.
* **Balanceo de carga inteligente**: Distribuye uniformemente el tráfico entre las instancias para evitar sobrecargas.
* **Integración con servicios de AWS**: Se integra con Route 53, Certificate Manager y CloudWatch.

Página oficial: [AMAZON ELB](https://aws.amazon.com/es/elasticloadbalancing/?nc=sn&loc=1)

Servicio de Almacenamiento Simple (S3)

El Servicio de Almacenamiento Simple (S3) de Amazon sería una excelente opción para almacenar los archivos de imágenes que nuestra aplicación necesitará subir. Este servicio nos proporciona almacenamiento por gigabyte (GB) a un costo principal de 0.02 USD por GB al mes, además de facturar la cantidad de solicitudes GET y PUT que realicemos.

Imagen que contiene Escala de tiempo

Descripción generada automáticamente

Estimación de costes

Se estima un gasto de:

* 2 instancias EC2 **(t4g.medium)** = 44.09 USD / mes
  + 2 vCPU
  + 4 GiB de RAM
  + Hasta 5 Gigabit de conexión
* 1 instancia RDS para PostgreSQL **(db.t4g.medium)** = 109.79 USD / mes
  + 2 vCPU
  + 4GiB de memoria RAM
  + 30GB de almacenamiento SSD.
* 1 instancia DocumentDB **(db.t4g.medium)** = 109.79 USD / mes
  + 2 vCPU
  + 4GiB de memoria RAM
  + 30GB de almacenamiento SSD.
* 1 instancia Elastic Load Balancing = 47,60 USD /mes
  + 2 GB por hora
* 1 instancia S3 = 11,51 USD / mes
  + 200GB de almacenamiento
  + 100000 solicitudes PUT, COPY, POST y LIST
  + 200000 solicitudes GET y SELECT

**Total 296,33€ / Mes**

\* Los costos mencionados son una estimación para un uso moderado de la aplicación. Amazon factura según el consumo real, por lo que los costos podrían ser más altos si la aplicación tiene una demanda mayor.

\* No se incluye el precio que agrega AWS de 0,05 USD / GB al mes en los datos de salida a Internet para ninguno de los servicios.

Costes de Producción

El sueldo mensual de cada uno de los trabajadores, habiendo durado el trabajo un mes, será de 1500€ aproximadamente.

En estos se incluyen las horas semanales invertidas en el proyecto, el uso de distintas herramientas de elaboración de código y la elaboración de tests unitarios y de integración.

También se incluyen las futuras tareas de mantenimiento y actualización que se llevarán a cabo a medida que vayan surgiendo distintos problemas.

Tabla

Descripción generada automáticamente

Elección de tecnologías

PHP con Laravel

**Popularidad y Comunidad Activa:** PHP es uno de los lenguajes de programación más utilizados en la web, con una gran comunidad de desarrolladores y muchos recursos disponibles en línea. Laravel, como uno de los frameworks PHP más populares, también tiene una comunidad activa y próspera que proporciona soporte, recursos educativos y paquetes adicionales que pueden acelerar el desarrollo.

**Facilidad de Aprendizaje:** PHP es conocido por ser relativamente fácil de aprender, especialmente para aquellos que ya tienen experiencia en programación. Laravel, construido sobre PHP, sigue este principio al proporcionar una curva de aprendizaje suave para los desarrolladores, gracias a su sintaxis clara y su amplia documentación.

**Arquitectura MVC (Modelo-Vista-Controlador):** Laravel sigue el patrón de diseño MVC, lo que promueve una separación clara de las preocupaciones en una aplicación web. Esto facilita la organización del código y mejora su mantenibilidad a medida que el proyecto crece.

**Rápido Desarrollo de Prototipos:** Laravel incluye muchas características que pueden acelerar el proceso de desarrollo, como el sistema de enrutamiento simple, la migración de bases de datos, el manejo de eventos y el soporte para tareas programadas. Estas características permiten a los desarrolladores crear prototipos rápidamente y llevar aplicaciones al mercado más rápidamente.

**Seguridad:** Laravel proporciona herramientas integradas para proteger las aplicaciones contra amenazas comunes, como la inyección SQL, la falsificación de solicitudes entre sitios (CSRF), y la inyección de código. Además, Laravel es compatible con la encriptación de contraseñas y la autenticación de dos factores, lo que ayuda a garantizar la seguridad de los datos de los usuarios.

**Ecosistema de Paquetes:** Laravel cuenta con un ecosistema robusto de paquetes y extensiones que pueden integrarse fácilmente en las aplicaciones para agregar funcionalidades adicionales. Esto permite a los desarrolladores aprovechar el trabajo de la comunidad y evitar tener que reinventar la rueda al construir nuevas características.

**Performance**: Aunque el rendimiento no suele ser la razón principal para elegir un framework, Laravel ha demostrado ser lo suficientemente eficiente en términos de rendimiento para la mayoría de las aplicaciones web. Además, Laravel proporciona herramientas para optimizar y cachear las aplicaciones, lo que puede mejorar aún más su rendimiento.

PostgreSQL

Hemos seleccionado PostgreSQL para gestionar los datos de empleados y clientes en nuestra aplicación debido a las siguientes razones:

Libre y de código abierto: PostgreSQL es una opción gratuita y de código abierto, lo que significa que no incurre en costos de licencia. Esta característica lo convierte en una elección atractiva para proyectos con recursos limitados, como startups y empresas emergentes.

Amplia comunidad y abundante documentación: PostgreSQL cuenta con una comunidad activa de desarrolladores y una extensa documentación. Esto garantiza que siempre haya recursos disponibles en línea, como guías, debates y tutoriales, lo que facilita tanto el aprendizaje como la resolución de problemas.

Adherencia a estándares de SQL: PostgreSQL sigue de cerca los estándares de SQL, lo que hace que aprender a utilizarlo sea altamente transferible a otras bases de datos relacionales. Esta consistencia en la sintaxis y la funcionalidad es beneficiosa para el desarrollo profesional a largo plazo.

Escalabilidad y rendimiento: PostgreSQL puede manejar eficientemente grandes volúmenes de datos a medida que nuestra aplicación crece. Su capacidad de escalar horizontal y verticalmente nos permite adaptarnos al crecimiento del proyecto sin sacrificar el rendimiento.

Funcionalidades avanzadas y extensibilidad: PostgreSQL ofrece una amplia gama de funcionalidades avanzadas, desde la gestión de datos espaciales hasta la compatibilidad con JSON y capacidades de indexación avanzada. Esto nos proporciona herramientas potentes para el manejo de datos y la optimización de consultas.

Seguridad sólida: La seguridad es una prioridad en PostgreSQL, ofreciendo mecanismos robustos de autenticación y autorización. Esto nos permite controlar con precisión quién accede a los datos y cómo lo hacen, protegiendo así la información confidencial de nuestros empleados y clientes.

En resumen, PostgreSQL se destaca como una opción de base de datos relacional por su gratuidad, su amplia comunidad y documentación, su adherencia a estándares de SQL, su escalabilidad, sus funcionalidades avanzadas y su sólida seguridad, convirtiéndolo en la elección ideal para almacenar los datos críticos de empleados y clientes en nuestra aplicación.

MongoDB

Hemos optado por utilizar MongoDB para gestionar los pedidos en nuestra aplicación por las siguientes razones:

Flexibilidad de esquema: MongoDB nos ofrece la libertad de guardar datos sin una estructura predefinida. Esto nos permite adaptarnos con facilidad a los cambios en nuestras aplicaciones, evitando la rigidez de las bases de datos relacionales.

Modelo de datos documental: Almacenar los datos en documentos similares a JSON simplifica enormemente su manipulación para los desarrolladores. Este modelo se alinea naturalmente con la estructura de datos común en muchas aplicaciones web modernas.

Escalabilidad horizontal sin complicaciones: MongoDB está diseñado para escalar horizontalmente sencillamente. Esta capacidad nos permite distribuir los datos en múltiples servidores, lo que resulta ideal para manejar grandes volúmenes de información y una mayor cantidad de usuarios sin sacrificar el rendimiento.

Eficiencia y velocidad: La optimización de MongoDB para un acceso rápido a los datos, junto con su capacidad de indexación, agiliza significativamente las consultas. Esto es fundamental para aplicaciones web que requieren respuestas rápidas y eficientes.

Amplia comunidad y recursos de aprendizaje: La activa comunidad de MongoDB y su documentación amigable nos proporcionan una gran cantidad de recursos en línea, como tutoriales y debates. Esto facilita tanto el aprendizaje como la resolución de problemas durante el desarrollo y la gestión de la aplicación.

Compatibilidad con datos no estructurados o semi-estructurados: MongoDB es una excelente elección cuando trabajamos con datos que no tienen un formato fijo o son semi-estructurados. Nos permite almacenar estos datos sin necesidad de ajustarlos a una estructura predefinida, lo que simplifica el proceso de almacenamiento y recuperación de la información.

En resumen, MongoDB destaca como una base de datos NoSQL que ofrece flexibilidad de esquema, eficiencia, escalabilidad horizontal y facilidad de uso para los desarrolladores, convirtiéndola en la elección ideal para gestionar los pedidos en nuestra aplicación.

En nuestro caso, usamos mongo para pedidos ya que es algo que va a tener una estructura flexible y nos facilita en el momento que queramos poder cambiar las propiedades de los pedidos o modificar algo en su estructura, a diferencia del resto de endpoints (productos, empleados, proveedores, categorías o clientes) que si van a tener una estructura más solida y definida y que además van a tener definidas relaciones claras entre ellos.

DIAGRAMA ENTIDAD RELACIÓN

