



GesMerCa





# GesMerCa

Gestión de  
recepcionamiento de  
mercancías con control  
de acceso





Estructura del código



Interfaz gráfica (GUI)



Flujo de la aplicación



Utilización de los usuarios

# La aplicación

Análisis y diseño



Estructura del código



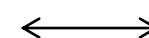
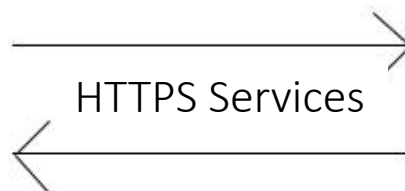
# Angular Architecture



User Interface (UI)

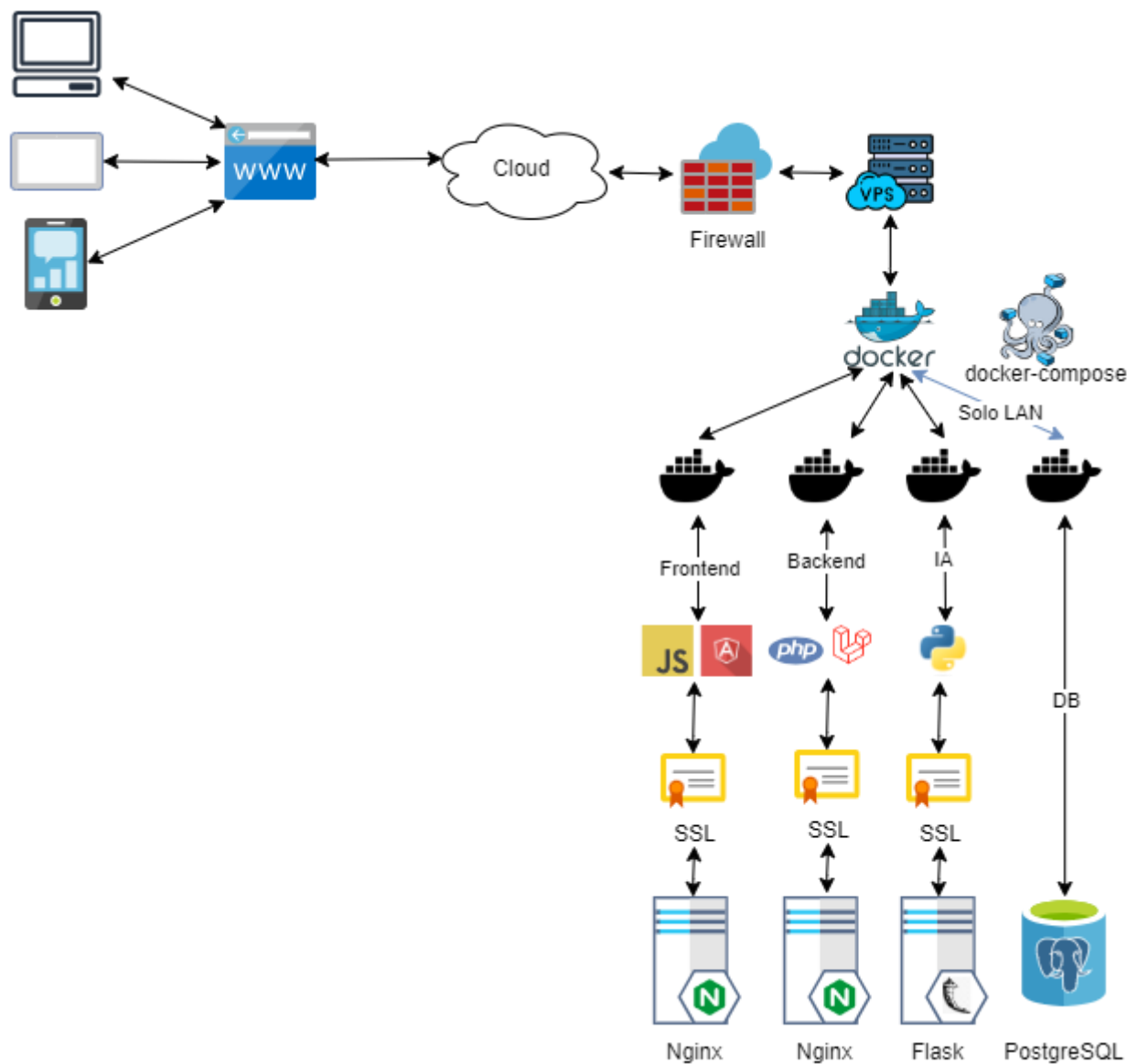


Data and Processing





Estructura del código

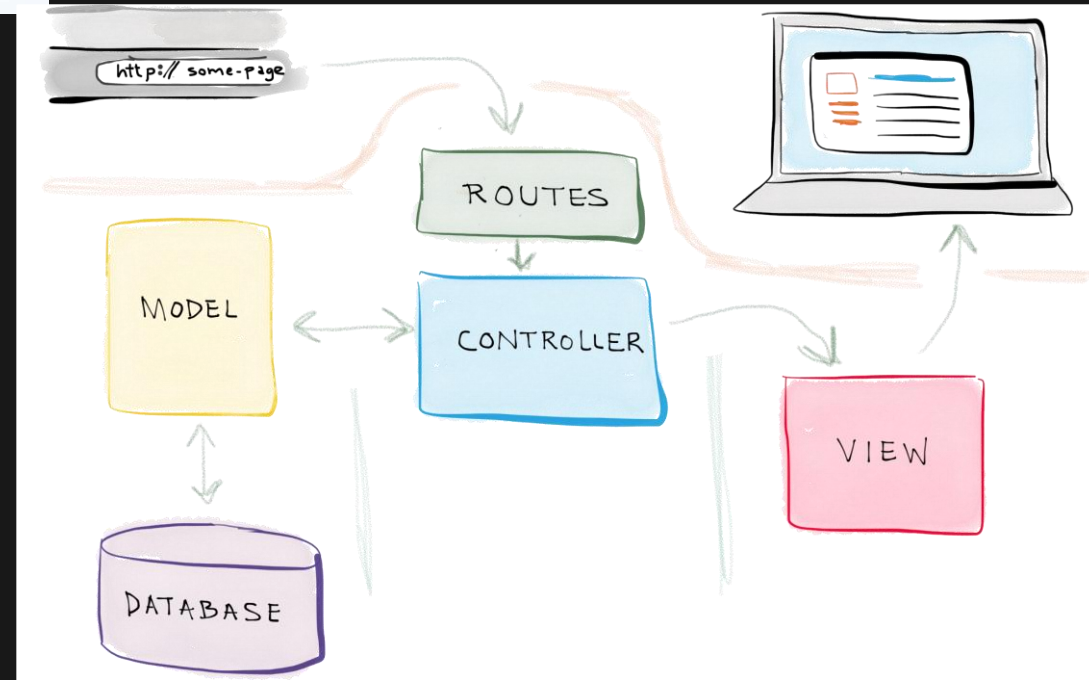
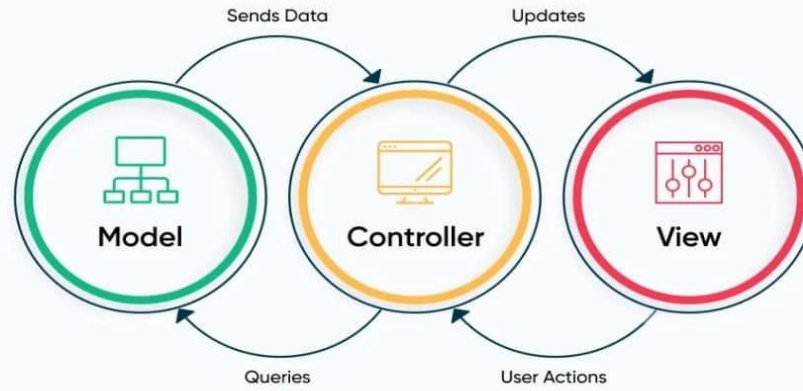




Estructura del código

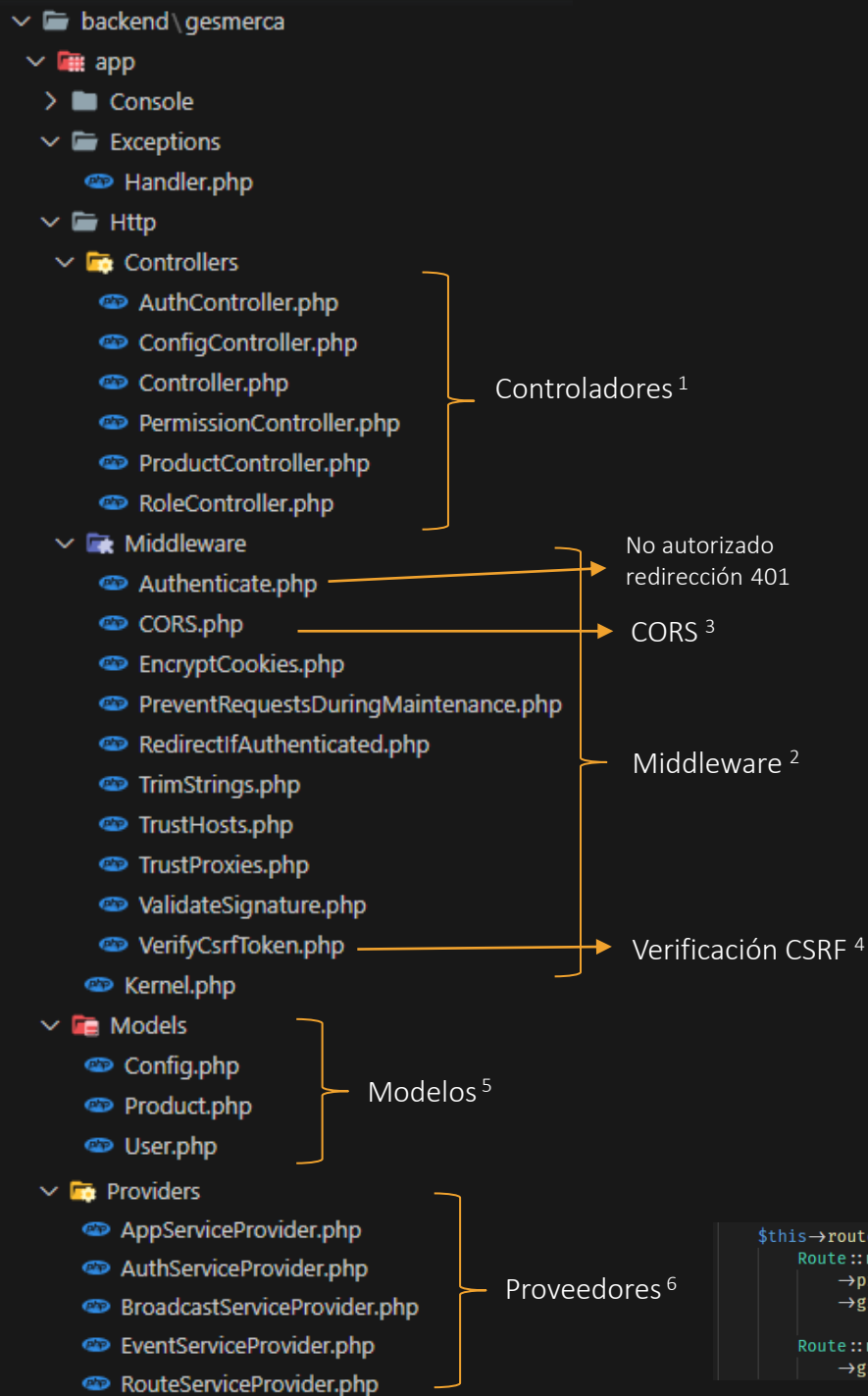


### Laravel MVC Architecture





## Estructura del código



(1) Contiene el código necesario para responder a las acciones que se solicitan en la aplicación, como visualizar un elemento, realizar una compra, una búsqueda de información, etc.

En realidad, es una capa que sirve de enlace entre las vistas y los modelos, respondiendo a los mecanismos que puedan requerirse para implementar las necesidades de nuestra aplicación. Sin embargo, su responsabilidad no es manipular directamente datos, ni mostrar ningún tipo de salida, sino servir de enlace entre los modelos y las vistas para implementar las diversas necesidades del desarrollo.

(2) El término middleware se refiere al sistema de software que ofrece funciones y servicios en la nube comunes para las aplicaciones, de modo que los desarrolladores y los equipos de operaciones puedan diseñarlas e implementarlas con mayor eficiencia. Además, permite conectar las aplicaciones, los datos y los usuarios.

(3) CORS se basa en un mecanismo por el cual los navegadores realizan una solicitud de "verificación previa" al servidor que aloja el recurso de origen cruzado, con el fin de comprobar que el servidor permitirá la solicitud real. En esa comprobación previa, el navegador envía cabeceras que indican el método HTTP y las cabeceras que se utilizarán en la solicitud real.

```
public function handle(Request $request, Closure $next) {
    header('Access-Control-Allow-Origin: *');
    header('Access-Control-Allow-Origin: Content-type, X-Auth-Token, Authorization, Origin');
    return $next($request);
}
```

(4) Los token CSRF permiten prevenir un frecuente agujero de seguridad de las aplicaciones web llamado "Cross Site Request Forgery". Básicamente es un ataque de seguridad que permite modificar el estado del servidor haciéndose pasar por un usuario determinado. El sitio web confía en el usuario pero la petición no es real y está siendo falsificada por el atacante.

(5) Son clases que representan a las tablas en la base de datos. Se utilizan para interactuar con la base de datos y realizar operaciones como insertar, actualizar y eliminar datos.

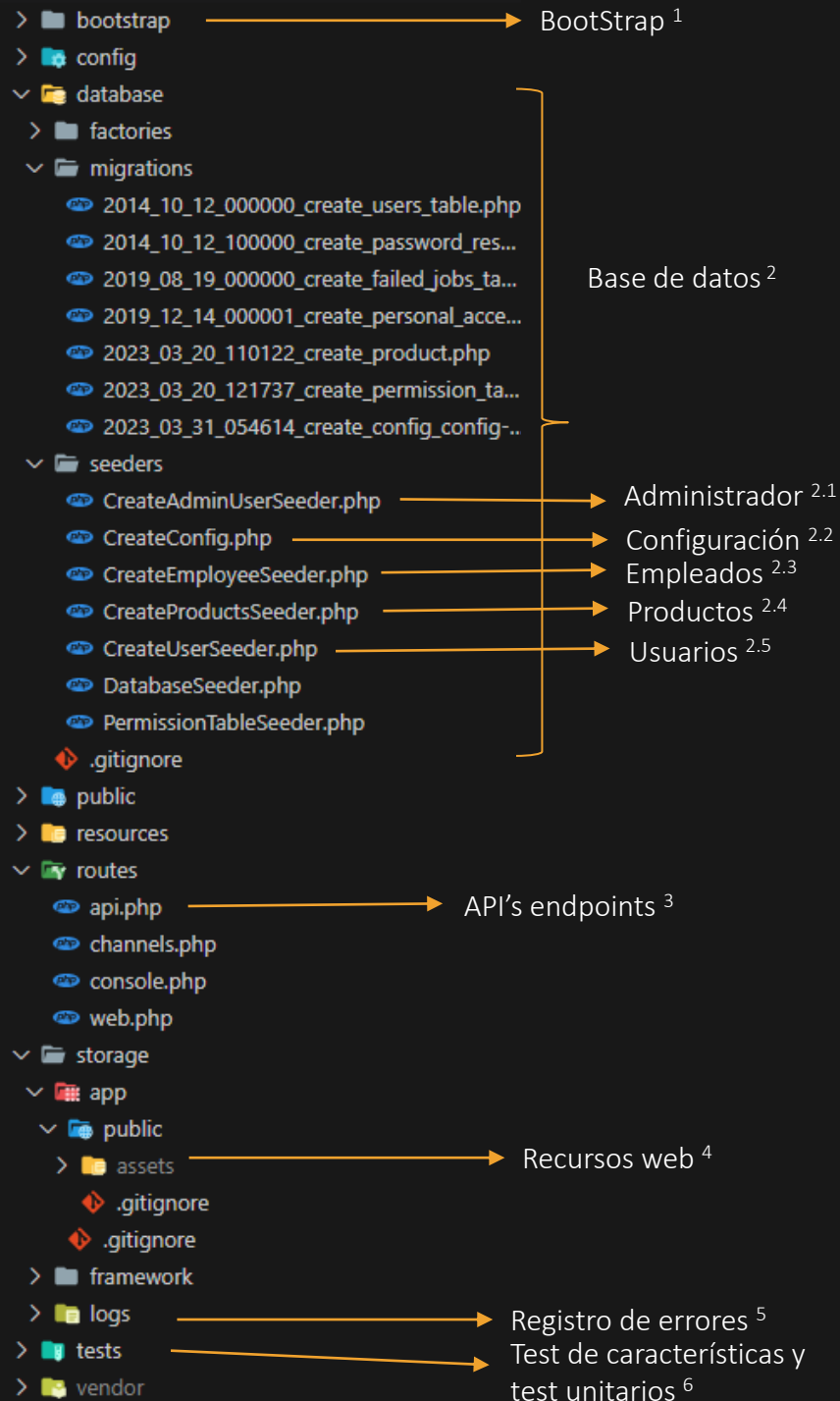
(6) Los proveedores de servicios son el lugar central de todo el arranque de aplicaciones de Laravel. Su propia aplicación, así como todos los servicios principales de Laravel, se inician a través de proveedores de servicios.

```
$this->routes(function () {
    Route::middleware('api')
        ->prefix('api')
        ->group(base_path('routes/api.php'));

    Route::middleware('web')
        ->group(base_path('routes/web.php'));
});
```

```
/**
 * Configure the rate limiters for the application.
 */
protected function configureRateLimiting(): void
{
    RateLimiter::for('api', function (Request $request) {
        return Limit::perMinute(60)->by($request->user()?->id ?: $request->ip());
    });
}
```

## Estructura del código



(1) Bootstrap es un framework front-end utilizado para desarrollar aplicaciones web y sitios mobile first, o sea, con un layout que se adapta a la pantalla del dispositivo utilizado por el usuario.

(2) Contiene las migraciones de la base de datos, las fábricas de modelos y las semillas. La carpeta migrations contiene la creación de las tablas y sus relaciones. La carpeta seeders contiene los juegos de datos de ejemplo que genera al realizar la migración de la base de datos.

(2.1) Crea el usuario administrador con el rol 'Administrador' y todos los permisos asignados.

(2.2) Crea la configuración general de la aplicación con la parametrización por defecto.

(2.3) Crea el usuario con el rol 'Empleado' y los permisos que le corresponden.

(2.4) Crea productos de ejemplo en la BBDD y crea imágenes para cada producto de forma aleatoria.

(2.5) Crea el usuario con el rol 'Usuario' y los permisos que le corresponden.

(3) Almacena y configura todos los endpoints de la aplicación

```
//Restricted access only product-list|product-create|product-edit|product-delete user permissions
Route::group([
    'middleware' => 'api',
    'prefix' => 'product'
], function ($router) {
    Route::get('/', [ProductController::class, 'index']);
    Route::get('/{id}', [ProductController::class, 'show'])->where('id', '[0-9]+');
    Route::post('/create', [ProductController::class, 'store']);
    Route::post('/delete', [ProductController::class, 'destroy']);
    Route::post('/update', [ProductController::class, 'update']);
});
```

(4) Contiene todos los recursos públicos de la web, es decir, las imágenes de los productos que se accederán directamente mediante el path público.

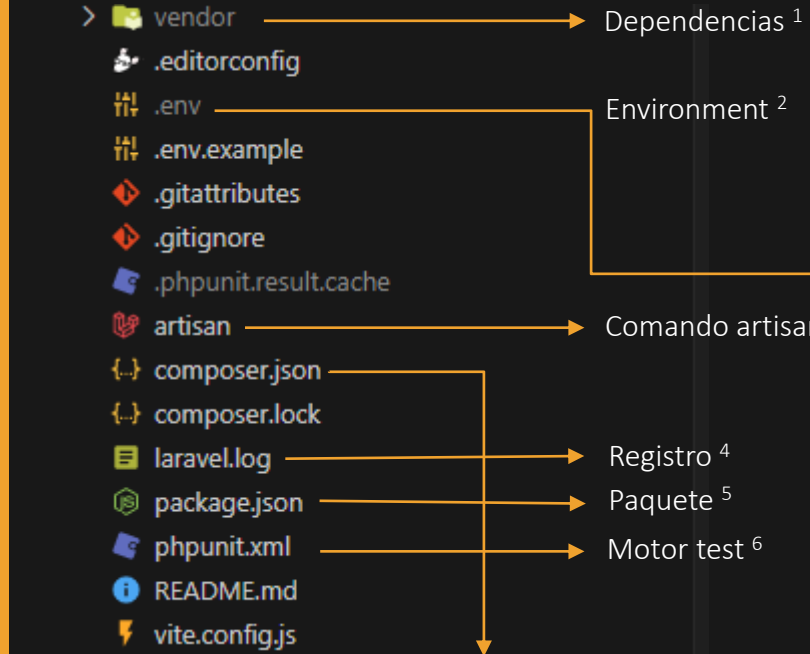
(5) Almacena cualquier error de 'debug' que se produzca durante la ejecución de la aplicación. El nivel se declara en el fichero .env de la aplicación.

(6) Contiene los scripts que ejecutará el proceso de testeo. El directorio contiene dos directorios: Feature y Unit. Las pruebas unitarias son pruebas que se enfocan en una porción muy pequeña y aislada de su código. De hecho, la mayoría de las pruebas unitarias probablemente se centren en un solo método. Las pruebas de características pueden probar una porción más grande de su código, incluida la forma en que varios objetos interactúan entre sí o incluso una solicitud HTTP completa a un punto final JSON.





## Estructura del código



```
{
  "name": "laravel/laravel",
  "type": "project",
  "description": "The Laravel Framework.",
  "keywords": ["framework", "laravel"],
  "license": "MIT",
  "require": {
    "php": "^8.1",
    "guzzlehttp/guzzle": "^7.2",
    "laravel/framework": "^10.0",
    "laravel/sanctum": "^3.2",
    "laravel/tinker": "^2.8",
    "spatie/laravel-permission": "^5.9",
    "tylon/jwt-auth": "^2.0"
  },
  "require-dev": {
    "fakerphp/faker": "^1.9.1",
    "laravel/pint": "^1.0",
    "laravel/sail": "^1.18",
    "mockery/mockery": "^1.4.4",
    "nunomaduro/collision": "^7.0",
    "phpunit/phpunit": "^10.0",
    "spatie/laravel-ignition": "^2.0"
  },
  "autoload": {
    "psr-4": {
      "App\\": "app/",
      "Database\\Factories\\": "database/factories/",
      "Database\\Seeders\\": "database/seeders/"
    }
  },
  "autoload-dev": {
    "psr-4": {
      "Tests\\": "tests/"
    }
  }
}
```

Dependencias de la aplicación <sup>7</sup>  
Roles y permisos  
Autenticación JWT

(7) Se ha intentado no instalar ningún paquete, ni dependencia adicional a la instalación por defecto de Laravel. Solamente se han instalado 2 paquetes (JWT y Roles y permisos) para evitar así problemas en futuros mantenimientos y actualizaciones.

- (1) Contiene todas las dependencias de la aplicación, ejecutando 'npm install' se descargan e instalan.
- (2) Fichero de variables de entorno, contiene toda la parametrización de la aplicación:

```
APP_NAME=GesMerCa
APP_ENV=local
APP_KEY=
APP_DEBUG=true
APP_URL=http://localhost

LOG_CHANNEL=stack
LOG_DEPRECATIONS_CHANNEL=null
LOG_LEVEL=debug

DB_CONNECTION=pgsql
DB_HOST=192.168.1.254
DB_PORT=5432
DB_DATABASE=
DB_USERNAME=
DB_PASSWORD=

BROADCAST_DRIVER=log
CACHE_DRIVER=file
FILESYSTEM_DISK=local
QUEUE_CONNECTION=sync
SESSION_DRIVER=file
SESSION_LIFETIME=120

MEMCACHED_HOST=127.0.0.1

REDIS_HOST=127.0.0.1
REDIS_PASSWORD=null
REDIS_PORT=6379

MAIL_MAILER=smt
MAIL_HOST=mailpit
MAIL_PORT=1025
MAIL_USERNAME=null
MAIL_PASSWORD=null
MAIL_ENCRYPTION=null
MAIL_FROM_ADDRESS="hello@example.com"
MAIL_FROM_NAME="${APP_NAME}"

AWS_ACCESS_KEY_ID=
AWS_SECRET_ACCESS_KEY=
AWS_DEFAULT_REGION=us-east-1
AWS_BUCKET=
AWS_USE_PATH_STYLE_ENDPOINT=false

PUSHER_APP_ID=
PUSHER_APP_KEY=
PUSHER_APP_SECRET=
PUSHER_HOST=
PUSHER_PORT=443
PUSHER_SCHEME=https
PUSHER_APP_CLUSTER=mt1

VITE_PUSHER_APP_KEY="${PUSHER_APP_KEY}"
VITE_PUSHER_HOST="${PUSHER_HOST}"
VITE_PUSHER_PORT="${PUSHER_PORT}"
VITE_PUSHER_SCHEME="${PUSHER_SCHEME}"
VITE_PUSHER_APP_CLUSTER="${PUSHER_APP_CLUSTER}"

JWT_SECRET=
```

Hash de la aplicación, se genera en cada compilación para comprobar la integridad  
Dominio de acceso a la aplicación

Parametrización de la base de datos

Parametrización del servidor de correo electrónico. En mi caso postfix.

Hash para la generación del token bearer

- (3) Artisan es la interfaz de línea de comandos incluida con Laravel.
- (4) Registro del log de los errores que se producen en la ejecución de los comandos, ej: en el php artisan.
- (5) Configuración de dependencias internas de laravel.
- (6) Motor de ejecución de los test unitarios de laravel.

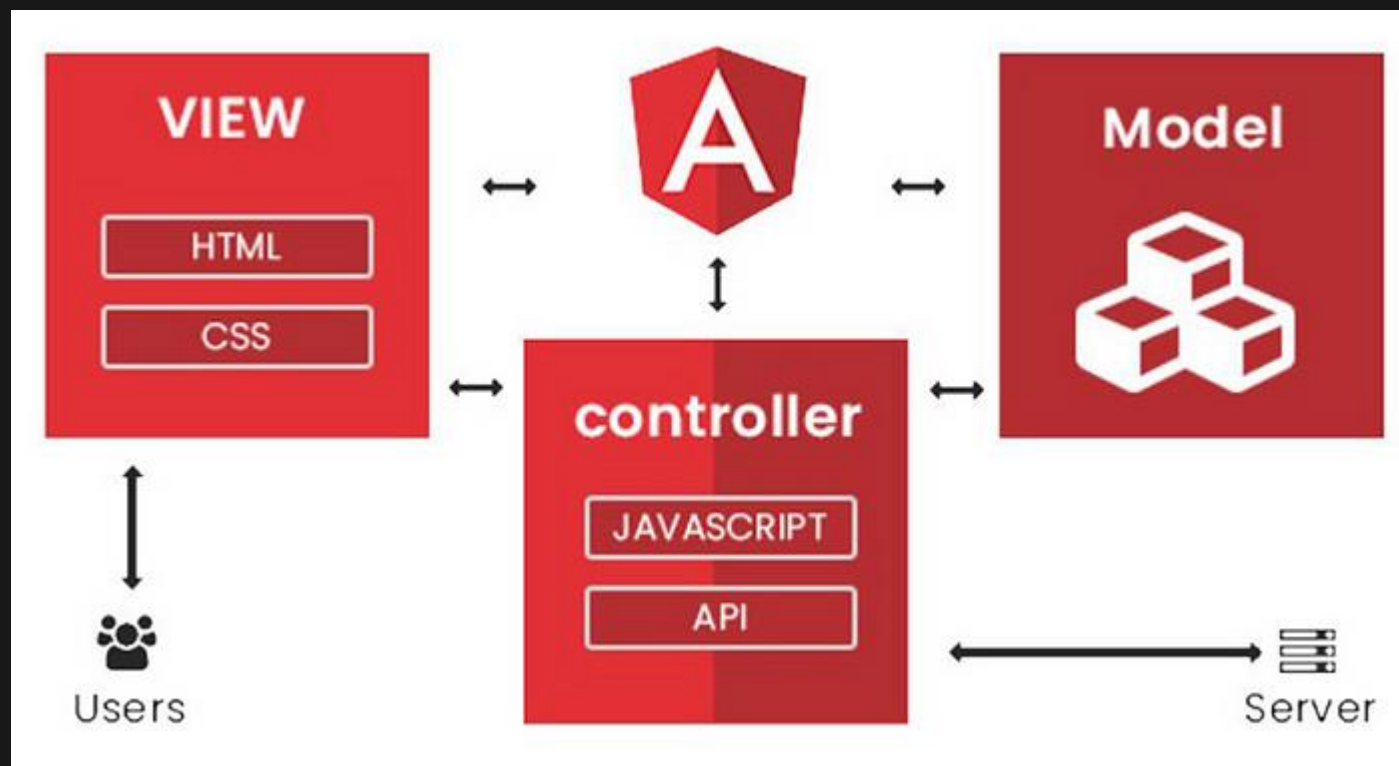


# Demostración

Código fuente Laravel



Estructura del código





## Estructura del código



(1) Al ejecutar el comando 'ng build' se transpila la aplicación y genera una versión minificada ubicada aquí. Es la versión enviada al servidor de producción.

(2) En Angular, un módulo es un mecanismo para agrupar componentes, directivas, conductos y servicios que están relacionados, de tal forma que se pueden combinar con otros módulos para crear una aplicación.

(2.1) Autenticación, está compuesto del: login, perfil y registro

(2.2) Configuración, está compuesto por: configuración general, configuración del usuario y edición de la configuración general

(2.3) Ayuda, es una ventana popup que se puede ubicar en cualquier punto de la aplicación. Contiene un video con las explicaciones de funcionamiento de la página en que se encuentra, capturado su nombre de forma dinámica por la ruta URI del navegador.

(2.4) Página de inicio de la aplicación, contiene la barra de tareas superior y un listado de todos los productos de la aplicación de forma paginada.

(2.5) Página de error 404, mostrada en error en la URI no existente.

(2.6) Permisos, lista con los permisos de cada usuario, permite la edición de la asignación de los permisos.

(2.7) Roles, lista con los roles de cada usuario, permite la edición de la asignación de los roles al usuario.

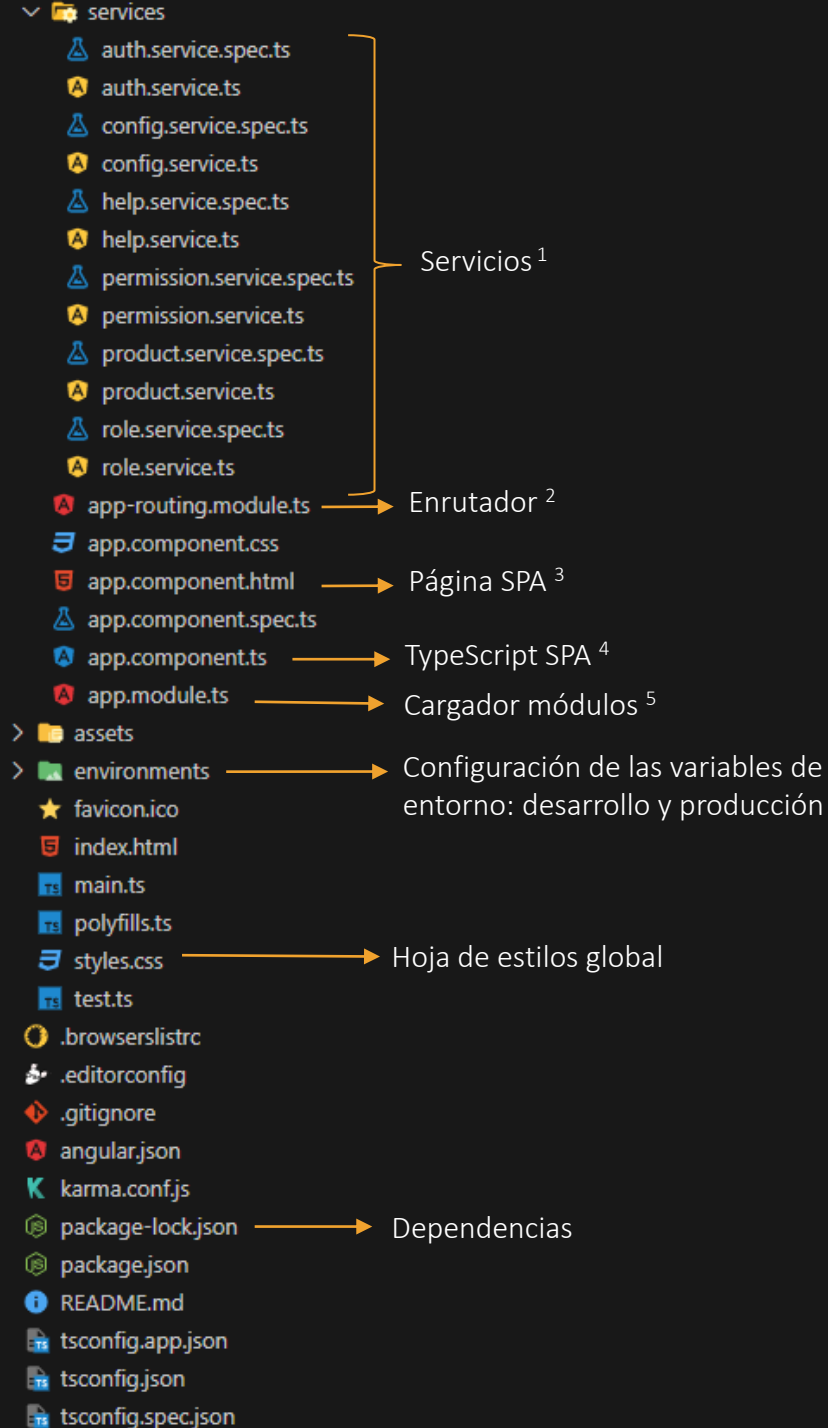
(3) Los protectores de ruta de Angular son interfaces proporcionadas por Angular que, cuando se implementan, nos permiten controlar la accesibilidad de una ruta en función de las condiciones proporcionadas en la implementación de clase de esa interfaz .

(4) Si ha trabajado antes con un servicio en Angular, los interceptores le sonarán familiares porque son un servicio Angular con implementación de interfaz `HttpInterceptor` . Los interceptores funcionan entre nuestra aplicación y el servidor e interactúan con la solicitud y la respuesta.

(5) Un modelo en angular es una clase que representa un objeto con propiedades que permite la utilización de la misma desde clases externas. Estos modelos de angular son interfaces que coinciden con los modelos recibidos mediante JSON de laravel, que a su vez coinciden en la mayoría de los casos con los modelos de Laravel, aunque puede disponer de una estructura o campos diferentes que se ha modificado por comodidad en el consumo de la API.

(\*) Los ficheros \*.spec.ts son ficheros de test unitario, correspondiente al typescript del mismo nombre.

## Estructura del código



(1) Los servicios son objetos que se instancian solo una vez durante la vida útil de una aplicación. Contienen métodos que mantienen los datos a lo largo de la vida de una aplicación, es decir, los datos están disponibles todo el tiempo. Es aquí donde se realiza el consumo de la API mediante observable al endpoint:

```
export class RoleService {
  baseUrl = environment.baseUrl;

  constructor(private http: HttpClient) {}

  getAll() {
    return this.http.get<Role[]>(`${this.baseUrl}/role/`);
  }

  getById(id: any) {
    return this.http.get<Role>(`${this.baseUrl}/role/${id}`);
  }

  getRoleUser(id: any) {
    return this.http.get<Role>(`${this.baseUrl}/role/user/${id}`);
  }

  setRoleUser(params: any, id: any) {
    return this.http.post(`${this.baseUrl}/role/user/${id}`, params);
  }

  create(params: any) {
    return this.http.post(`${this.baseUrl}/role/create`, params);
  }

  update(params: any, id: any) {
    return this.http.post(`${this.baseUrl}/role/update/${id}`, params);
  }

  delete(params: any, id: any) {
    return this.http.post(`${this.baseUrl}/role/delete/${id}`, params);
  }
}
```

(2) El enrutamiento en Angular permite a los usuarios crear una aplicación de una sola página con múltiples vistas y permite la navegación entre ellas. Los usuarios pueden cambiar entre estas vistas sin perder el estado y las propiedades de la aplicación.

(3-4) Es un tipo de aplicación web donde todas las pantallas las muestra en la misma página, sin recargar el navegador. Éste es el punto de entrada, el equivalente al `index.html`.

(4) El cargador angular permite que sus scripts angulares se carguen en cualquier orden.

(\*) Los archivos `*.spec.ts` son ficheros de test unitario, correspondiente al typescript del mismo nombre.



# Demostración

Código fuente Angular



Estructura del código



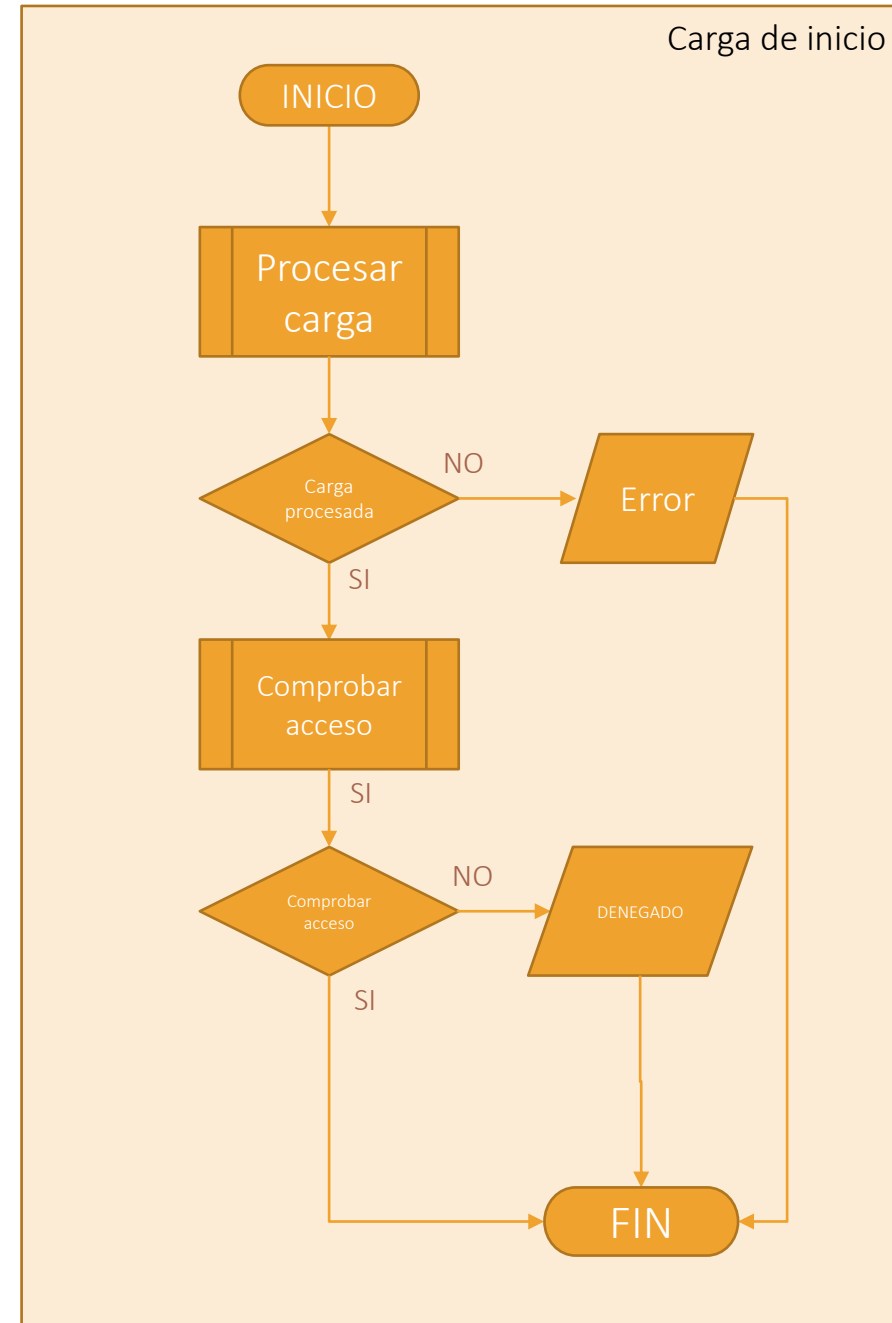
Interfaz gráfica (GUI)



Flujo de la aplicación



Utilización de los usuarios





Estructura del código



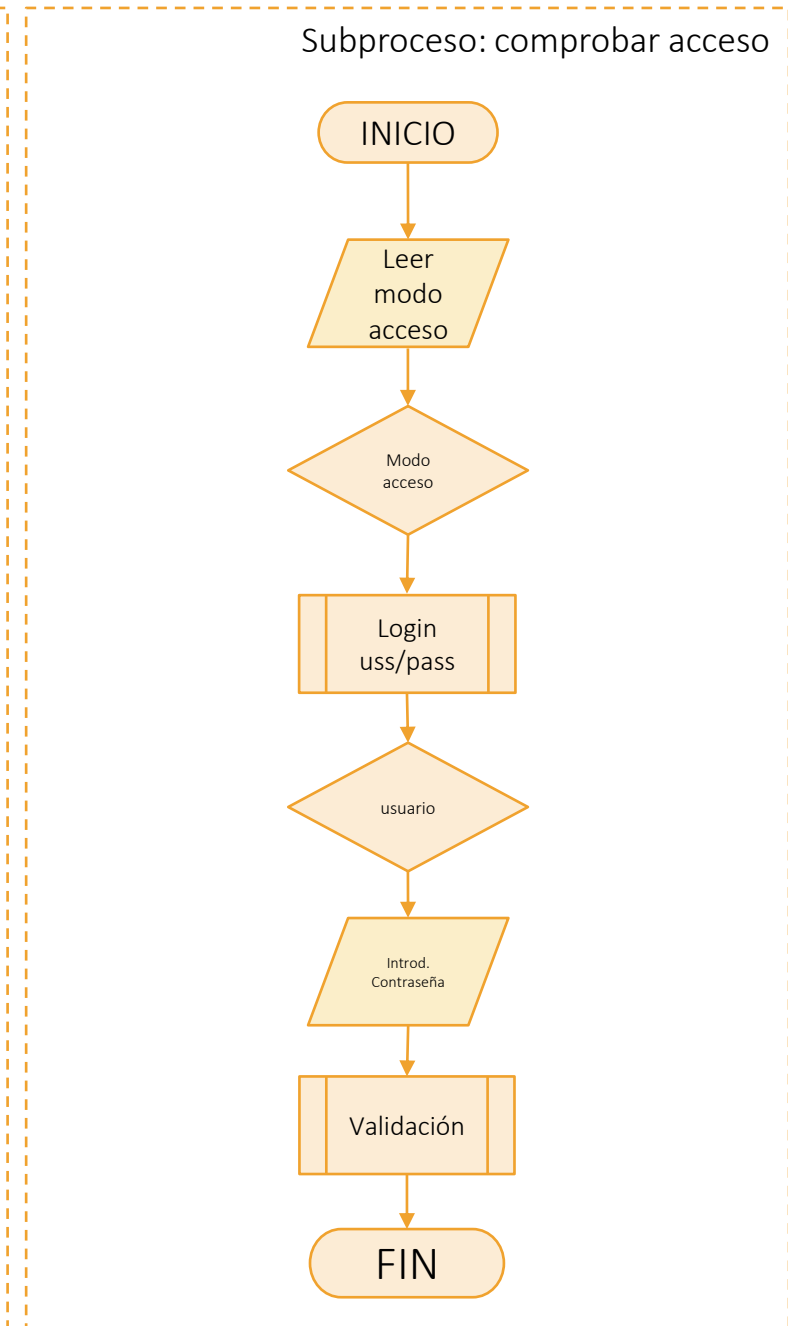
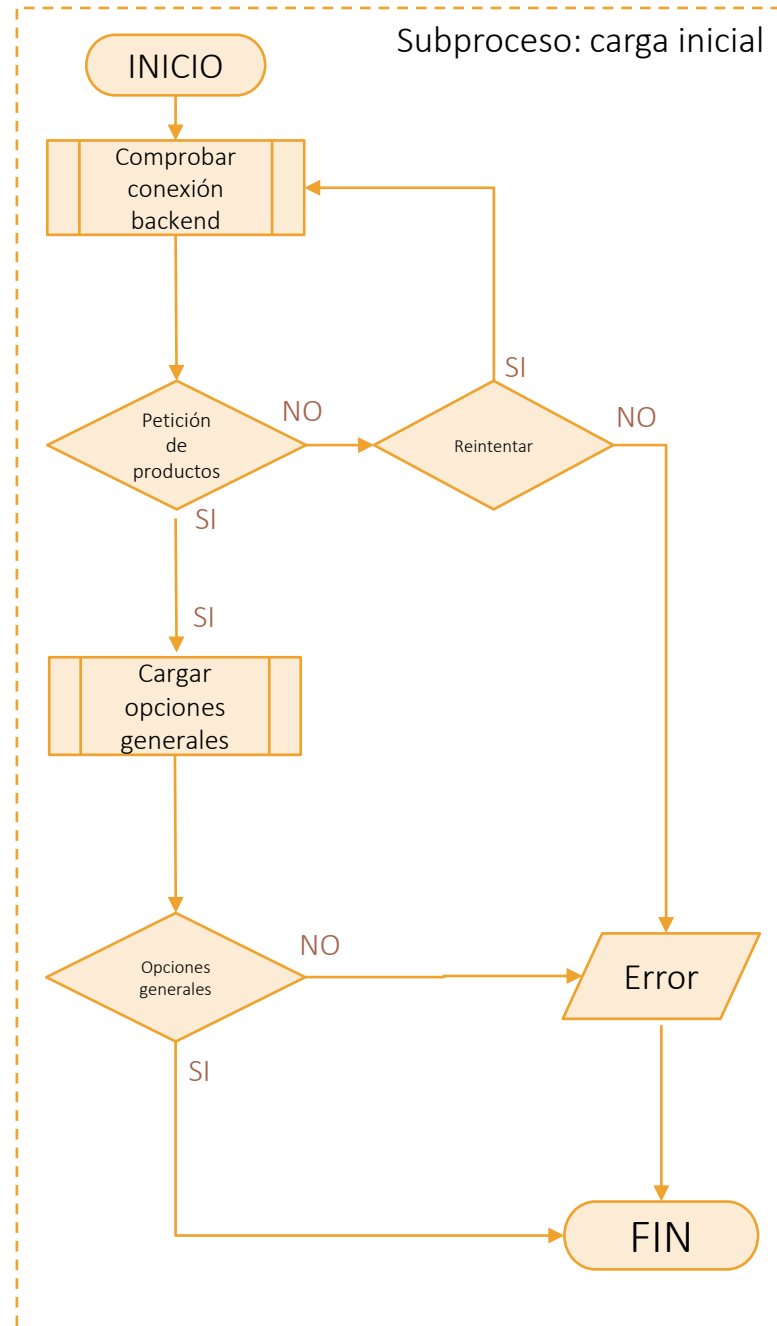
Interfaz gráfica (GUI)



Flujo de la aplicación



Utilización de los usuarios







Estructura del código



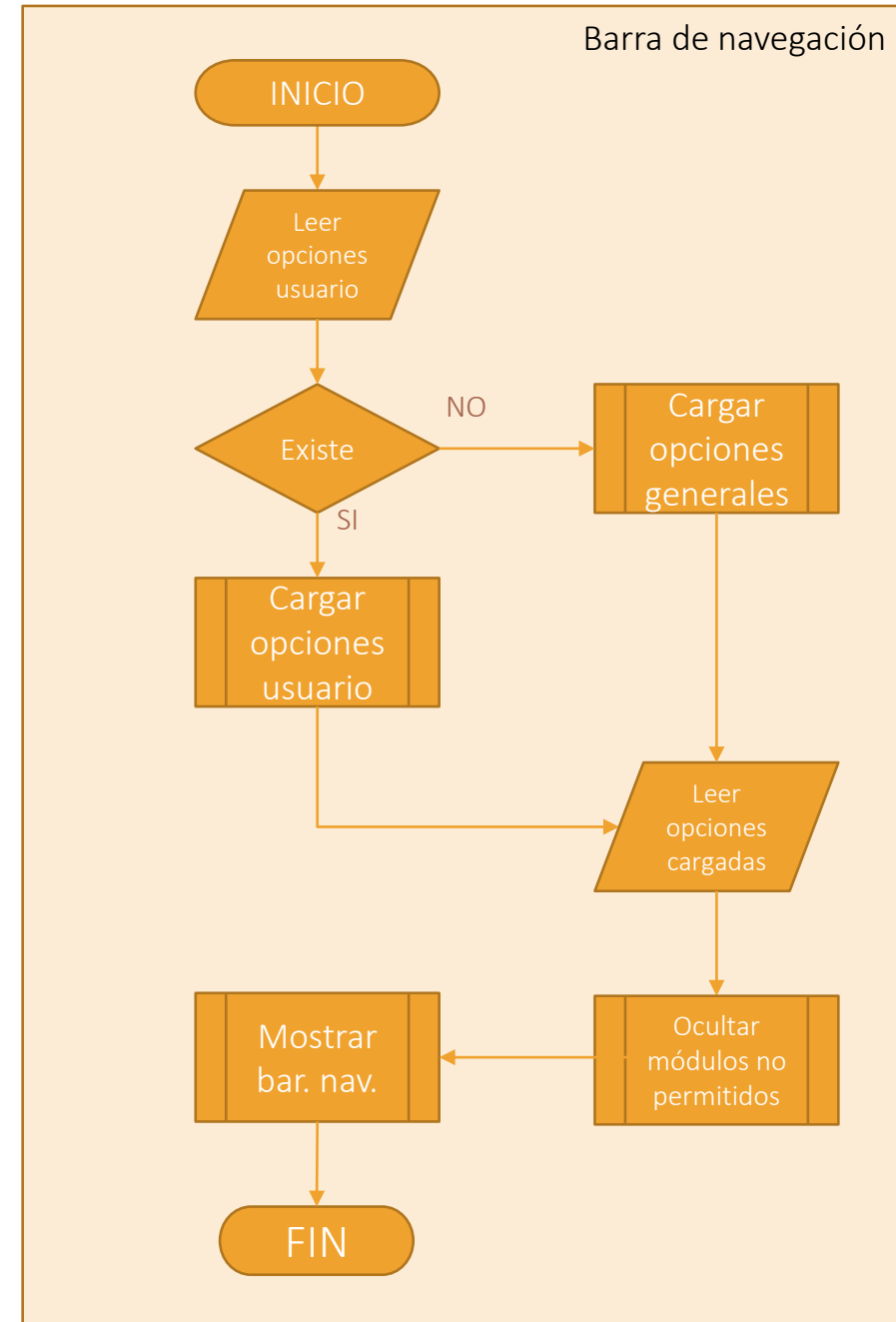
Interfaz gráfica (GUI)



Flujo de la aplicación



Utilización de los usuarios





Estructura del código



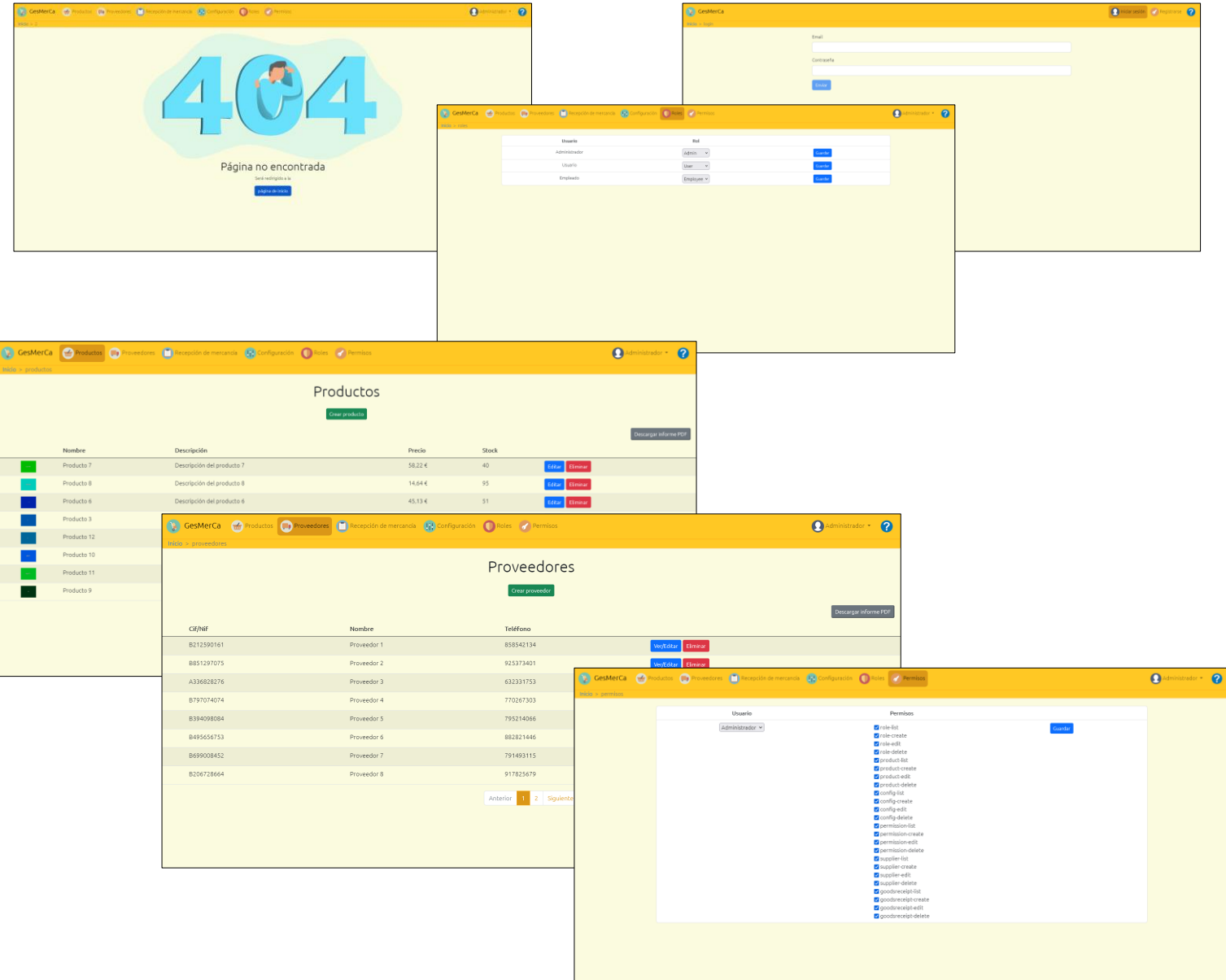
Interfaz gráfica (GUI)



Flujo de la aplicación



Utilización de los usuarios





Estructura del código



Interfaz gráfica (GUI)



Flujo de la aplicación



Utilización de los usuarios

## El diseño de interfaces de los formularios

Se ha basado en la **clasificación según la funcionalidad** final hacia el usuario.

- **Colores.** Para cada módulo y funcionalidad se ha establecido un color determinado.
- **Posición.** En la barra de tareas se ha agrupado las funcionalidades más habituales según roles y asiduidad.
- **Iconos.** Se han utilizado iconos de identificación en base a la funcionalidad y teniendo en cuenta tanto el color principal como el de fondo.

## Diseño flat o diseño plano

Consiste en eliminar o reducir todo tipo de decoración en un diseño de interfaz para simplificar el mensaje y facilitar la funcionalidad.

Se eliminan texturas, degradados, biselados, sombreados... en definitiva, todo lo que no aporte valor al mensaje o información que se quiere transmitir al usuario que interactúa con la interfaz.

- **Colores:** se suelen utilizar colores vivos en paletas pastel con pocas variaciones de cada color.
- **Tipografía:** se suelen utilizar grandes tamaños, al mismo tiempo que se reducen las longitudes de los títulos y subtítulos.
- **Mensaje:** el mensaje suele ir reforzado con códigos de color para dar jerarquía o importancia a la información.





Estructura del código



Interfaz gráfica (GUI)

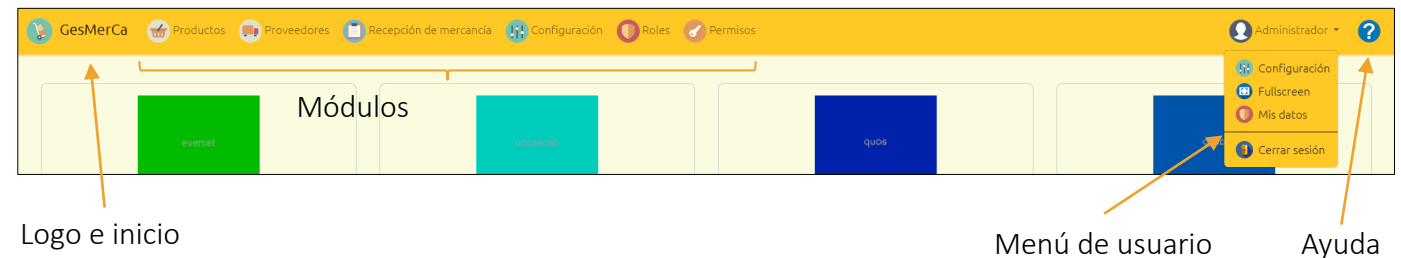
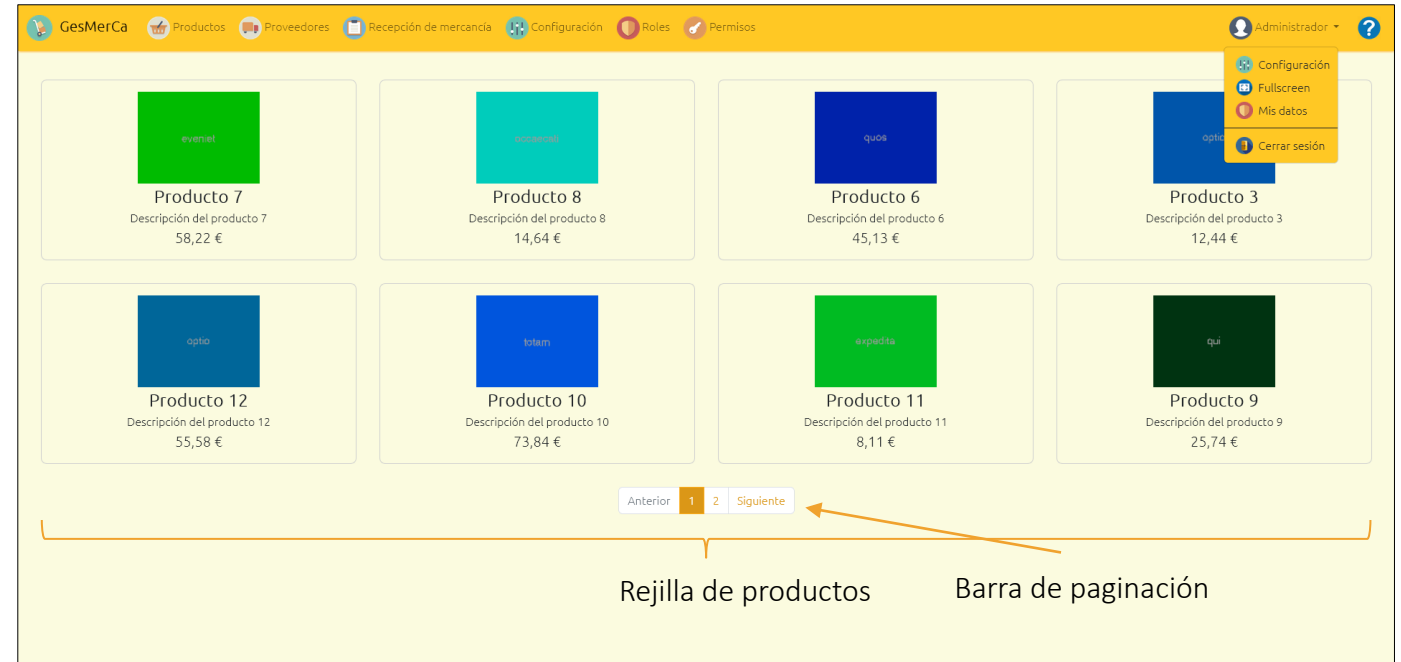


Flujo de la aplicación



Utilización de los usuarios

## El diseño de inicio y la barra de navegación





Estructura del código



Interfaz gráfica (GUI)



Flujo de la aplicación



Utilización de los usuarios

## El diseño de los listados

GeMerCa | Productos | Proveedores | Recepción de mercancía | Configuración | Roles | Permisos | Administrador

Inicio > proveedores

Breadcrumb

Listado

Proveedores

Crear proveedor

Descarga de informes

Descargar informe PDF

Alta de elementos

Cif/Nif	Nombre	Teléfono	
B212590161	Proveedor 1	858542134	Ver/Editar Eliminar
B851297075	Proveedor 2	925373401	Ver/Editar Eliminar
A336828276	Proveedor 3	632331753	Ver/Editar Eliminar
B797074074	Proveedor 4	770267303	Ver/Editar Eliminar
B394098084	Proveedor 5	795214066	Ver/Editar Eliminar
B495656753	Proveedor 6	882821446	Ver/Editar Eliminar
B699008452	Proveedor 7	791493115	Ver/Editar Eliminar
B206728664	Proveedor 8	917825679	Ver/Editar Eliminar

Botones de modificación y eliminación

Anterior 1 2 Siguiente

Barra de paginación

GeMerCa | Productos | Proveedores | Recepción de mercancía | Configuración | Roles | Permisos | Administrador

Inicio > productos

Productos

Crear producto

Descargar informe PDF

Numero	Descripción	Precio	Stock	
Producto 7	Descripción del producto 7	58,22 €	40	Ver/Editar Eliminar
Producto 8	Descripción del producto 8	14,64 €	95	Ver/Editar Eliminar
Producto 6	Descripción del producto 6	45,13 €	51	Ver/Editar Eliminar
Producto 3	Descripción del producto 3	12,44 €	10	Ver/Editar Eliminar
Producto 12	Descripción del producto 12	55,58 €	65	Ver/Editar Eliminar
Producto 10	Descripción del producto 10	73,84 €	85	Ver/Editar Eliminar
Producto 11	Descripción del producto 11	8,11 €	25	Ver/Editar Eliminar
Producto 9	Descripción del producto 9	23,74 €	65	Ver/Editar Eliminar

Anterior 1 2 Siguiente

GeMerCa | Productos | Proveedores | Recepción de mercancía | Configuración | Roles | Permisos | Administrador

Inicio > recepción de mercancía

Recepción de mercancía

Crear alta de recepción de mercancía

Descargar informe PDF

Núm. docum.	Proveedor	Usuario	Fecha	
B753623724	Proveedor 7	Usuario	09/08/2023	Ver/Editar Eliminar
B473587333	Proveedor 9	Administrador	29/07/2023	Ver/Editar Eliminar
B33687427	Proveedor 11	Usuario	27/07/2023	Ver/Editar Eliminar
B180231387	Proveedor 14	Administrador	21/07/2023	Ver/Editar Eliminar
A355307382	Proveedor 14	Usuario	28/05/2023	Ver/Editar Eliminar
B387915087	Proveedor 2	Administrador	14/05/2023	Ver/Editar Eliminar
B97607406	Proveedor 4	Administrador	14/05/2023	Ver/Editar Eliminar
A482346290	Proveedor 11	Administrador	28/12/2022	Ver/Editar Eliminar

Anterior 1 2 3 4 5 6 7 8 9 10 11 12 Siguiente



Estructura del código



Interfaz gráfica (GUI)



Flujo de la aplicación



Utilización de los usuarios

## El diseño de los formularios de edición y visualización

Annotations in the screenshot:

- Breadcrumb: Inicio > productos > editar > 7
- Imagen: eveniet
- Campos del formulario: Nombre, Descripción, Proveedor, Imagen, Precio, Precio Mín., Precio Máx., Precio Medio, Stock.
- Botón de envío y validación: Enviar



Estructura del código



Interfaz gráfica (GUI)



Flujo de la aplicación



Utilización de los usuarios

## El diseño de los formularios de alta

Inicio > productos > nuevo

Breadcrumb

Nombre

Descripción

Proveedor

Imagen

Examinar... No se ha seleccionado ningún archivo.

Precio

Stock

Crear producto

Campos del formulario

Imagen

Botón de envío y validación





Estructura del código



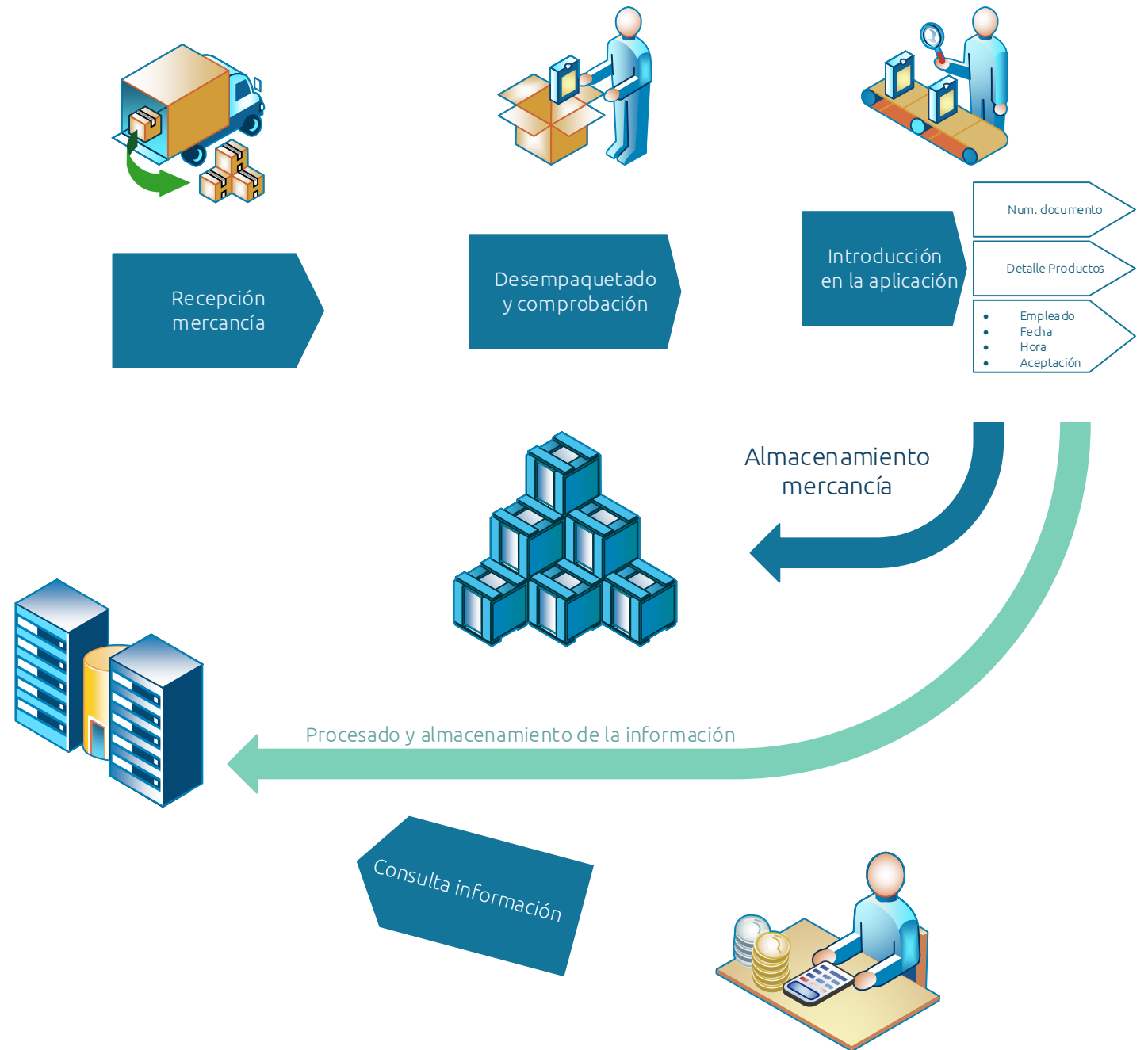
Interfaz gráfica (GUI)



Flujo de la aplicación



Utilización de los usuarios





Estructura del código



Interfaz gráfica (GUI)



Flujo de la aplicación



Utilización de los usuarios

## Text To Speech (TTS)

La síntesis de voz **es la producción artificial del habla humana**. Se han diseñado diferentes sistemas para este propósito llamados sintetizadores de voz y pueden ser implementados tanto en hardware como en software.

Un sistema *TTS* (*text to speech*) o sintetizador de voz **convierte el lenguaje escrito en habla**.

El habla sintetizada se genera concatenando **segmentos de grabaciones** que se encuentran **almacenados en una base de datos**.

Los sistemas *text to speech* difieren en diversos aspectos, uno de ellos es el tamaño de las unidades de habla almacenados. Los sistemas que almacenen fonemas y di fonemas proveen el rango de salida más amplio, sin embargo es posible que su calidad sea baja.

Para una salida de alta calidad, se utiliza la técnica de dominios específicos; en esta técnica el almacenamiento de palabras u oraciones pre-grabadas enteras permiten una salida de alta calidad.

Un sistema *TTS* puede lograr que **personas con discapacidad visual**, de lectura o en actividades en las cuales ojos y manos están ocupadas **puedan escuchar** instrucciones, textos leídos, entre otros textos.

Muchos sistemas operativos han incluido sintetizadores de voz desde principios de la década de los ochenta y su calidad ha variado mucho de modelo en modelo, incluso actualmente hay *TTS* de muy variados tipos y calidades.



Estructura del código



Interfaz gráfica (GUI)



Flujo de la aplicación



Utilización de los usuarios

## Alto contraste

El modo de alto contraste permite al usuario invertir los **colores de primer plano y de fondo**, lo que a menudo **ayuda** a que el texto se destaque mejor. Para alguien con una discapacidad de baja visión, el modo de alto contraste puede **facilitar la navegación** por el contenido de la página.

La lista de verificación de WebAIM establece en la sección 1.4.1 que "**el color no debe usarse como el único método** para transmitir contenido o distinguir elementos visuales". También señala que "el color solo no debe usarse para distinguir los enlaces del texto circundante" a menos que cumplan **ciertos requisitos de contraste**. En cambio, la lista de verificación recomienda agregar un indicador adicional, como un guión bajo (usando la propiedad CSS text-decoration) para indicar cuándo el enlace está activo.

Hay aproximadamente 320 millones de personas en todo el mundo con deficiencia de visión de los colores. Aproximadamente 1 de cada 12 hombres y 1 de cada 200 mujeres tienen alguna forma de "daltonismo"; eso significa que alrededor de una vigésima parte, o el 5%, de sus usuarios no experimentarán su sitio de la manera deseada. Cuando confiamos en el color para transmitir información, llevamos ese número a niveles inaceptables.

*Nota: El término "daltonismo" se utiliza a menudo para describir una condición visual en la que una persona tiene problemas para distinguir los colores, pero de hecho, muy pocas personas son realmente daltónicas. La mayoría de las personas con deficiencias de visión de color pueden ver algunos o la mayoría de los colores, pero tienen dificultades para diferenciar algunos, como rojos y verdes (los más comunes), marrones y naranjas, y azules y púrpuras.*



# Demostración

Página inicial y barra de navegación



# Demostración

Alta de productos, proveedores  
y albarán de recepción de mercancía



# Demostración

Configuración general, usuarios y permisos



# GesMerCa

Gestión de  
recepcionamiento de  
mercancías con control  
de acceso

