

Universidad de Costa Rica

Facultad de Ingeniería

Escuela de Ciencias de la Computación

Seguridad de sistemas computacionales
CI-0143

Tarea (Parte II)

Desarrollo, seguridad y aseguramiento de una aplicación web.

Profesor:
Ricardo Villalón

Elaborado por:
Hellen Fernández Jiménez B42525
Stephanie María Leitón Ramirez B74106
Roberto Vargas Rojas B57617
Sergio Ortega Carpio B65162

Primer ciclo 2021

Índice

1. Componentes del servicio asegurar	2
2. Objetivos generales de la seguridad	2
3. Seguridad de la aplicación web	2
3.1. Máquina virtual	2
3.2. Servidor Web Apache	3
3.2.1. Configuración para conexiones HTTPS (Parcial)	3
3.3. MariaDB	5
3.4. Módulo Home	6
3.4.1. Objetivos de la seguridad	6
3.4.2. Interacciones	6
3.4.3. Vulnerabilidades y amenazas	7
3.4.4. Riesgos	7
3.5. Módulo de Login	7
3.5.1. Objetivos de la seguridad	7
3.5.2. Interacciones	7
3.5.3. Vulnerabilidades y amenazas	8
3.5.4. Riesgos	8
3.6. Módulo de Registro	9
3.6.1. Objetivos de la seguridad	9
3.6.2. Interacciones	9
3.6.3. Vulnerabilidades y amenazas	10
3.6.4. Riesgos	10
3.7. Módulo de Comentarios	11
3.7.1. Objetivos de la seguridad	11
3.7.2. Interacciones	11
3.7.3. Vulnerabilidades y amenazas	11
3.7.4. Riesgos	12
3.8. Módulo para agregar productos	12
3.8.1. Objetivos de la seguridad	12
3.8.2. Interacciones	13
3.8.3. Vulnerabilidades y amenazas	13
3.8.4. Riesgos	14
3.9. Módulo para listar productos	14
3.9.1. Objetivos de la seguridad	14
3.9.2. Interacciones	15
3.9.3. Vulnerabilidades y amenazas	16
3.9.4. Riesgos	16
3.10. Módulo para ver un carrito	17
3.10.1. Objetivos de la seguridad	17
3.10.2. Interacciones	17
3.10.3. Vulnerabilidades y amenazas	18
3.10.4. Riesgos	18
4. Políticas y controles de la aplicación web	19

1. Componentes del servicio asegurar

Los componentes a asegurar incluyen lo siguiente:

1. Máquina virtual.
2. Servicio del servidor web Apache.
3. Servicio de MariaDB para gestionar bases de datos.
4. La aplicación web construida, con todos sus componentes/módulos.

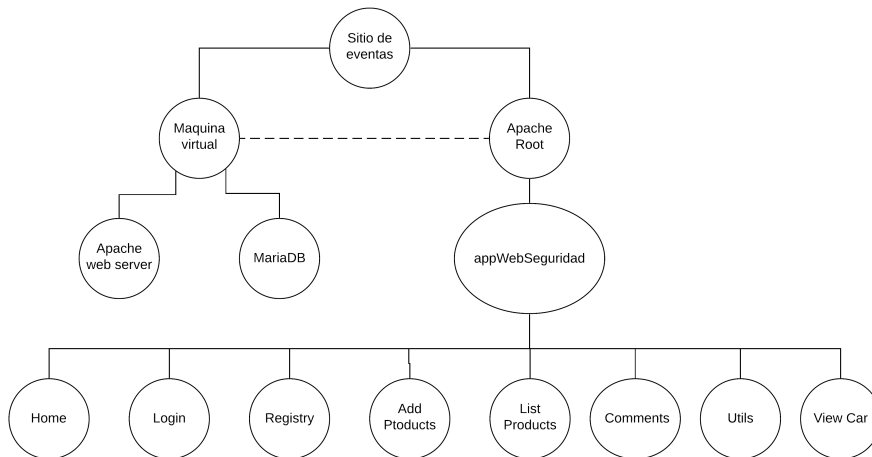


Figura 1: Árbol de componentes a asegurar

2. Objetivos generales de la seguridad

1. Mantener la confidencialidad de la información de las personas en el contexto del Sitio de ventas.
2. Autenticación de usuarios en su interacción con el sitio de ventas.
3. Autorización a los usuarios dentro del sitio de ventas para realizar ciertas acciones.
4. Integridad de la información de pago (tarjetas de crédito).
5. La integridad de la información (ingresada y existente) y de los componentes y/o sistemas que conforman el sitio de ventas.
6. Auditoría de eventos relevantes en el uso de la aplicación.

3. Seguridad de la aplicación web

3.1. Máquina virtual

- Para la seguridad de la máquina virtual se utiliza Selinux y la contraseña que fue otorgada al momento de reservar la computadora para el usuario hellen.fernandezjimenez.
- Adicionalmente se verificó que para los directorios de la aplicación `/var/www/cgi-bin` y `/var/www/cgi-bin/appWebSeguridad` los permisos para escribir y leer solo los tengan los dueños de los archivos (apache). Se evitó usar `chmod 777`.

3.2. Servidor Web Apache

Se realizó lo siguiente para asegurar la configuración de apache:

- Solo se abrieron los puertos necesarios en la máquina para que el apache funcione.
- El archivo **httpd.conf** se modificó para evitar que apache liste los directorios del servidor por medio del navegador.
- El archivo **httpd.conf** se modificó para prohibir la lectura por medio del navegador para **todo** tipo de archivo .h y .cpp que contenga la aplicación.
- El archivo httpd.conf se modificó para desactivar Sym Links.

3.2.1. Configuración para conexiones HTTPS (Parcial)

Como mecanismo de seguridad se decidió aplicar una configuración especial al servidor Apache para que fuese capaz de procesar solicitudes a través del protocolo HTTPS. Cabe recalcar que después de varios intentos de configuración, aún la aplicación no logra procesar la solicitud. Algunas características de la configuración realizada en el servidor CentOS, se listan a continuación:

- Primeramente se creó un certificado digital haciendo uso de la librería **OpenSSL**. Se debe crear un archivo de extensión .crt y otro .key. Ambos deben guardarse en carpetas específicas que son en las que confía el servidor Apache. Las ruta son **/etc/pki/tls/certs/** y **/etc/pki/tls/private/** respectivamente.
- Posteriormente, se debe verificar que el modulo de ssl se encuentre disponible dentro de los archivos de configuración del servidor. Esto puede ser verificado en la siguiente ruta **/etc/httpd/modules/mod_ssl.so**.
- Paralelo al punto anterior, debe existir un archivo de configuración que por defecto, contiene las configuraciones adicionales para poder escuchar por el puerto 443, que es el puerto estándar para las solicitudes seguras en la web. Se puede verificar la existencia de este archivo en la siguiente ruta **/etc/httpd/conf.d/ssl.conf**. El archivo básicamente hace uso de la cláusula VirtualHost para definir las propiedades requeridas por el servidor para SSL.
- Dentro del archivo anterior, se debe cambiar el nombre de los certificados que estan por defecto, por los valores de los certificados que fueron creados en pasos anteriores. Dichos archivos se deben encontrar en las mismas carpetas que vienen ya configuradas, no obstante, el certificado es el que debe ser cambiado.
- Las propiedades anteriores son agregadas al archivo de **httpd.conf**, que es el archivo general de configuración. La adición de la configuración SSL es por medio de un Include del path del punto anterior, de manera que cuando la configuración general cargue, esta cargue también el virtual host creado en **ssl.conf**.
- Finalmente, tra reiniciar el servidor, se comprueba por medio del siguiente comando, que el servidor efectivamente se encuentra escuchando por el puerto 443, sin embargo, al tratar de acceder al sitio por https, este no responde.

```
$ netstat -atun
```

```
tcp6      0      0 :::443          :::*           LISTEN
tcp6      0      0 :::80           :::*           LISTEN
```

Como se menciona anteriormente, el error presentado parece ser ocasionado por los certificados. El servidor Apache cuenta con archivos de bitácora dónde se pueden consultar los errores relacionados a ssl. Dichos logs, pueden ser consultados en la siguiente ruta */etc/httpd/logs/ssl_error_log*.

```
[Wed Jun 02 20:09:08.328016 2021] [ssl:warn] [pid 77430] AH01906: RSA server certificate is a CA certificate (BasicConstraints: CA == TRUE !?)
[Wed Jun 02 20:09:08.352039 2021] [ssl:warn] [pid 77430] AH01906: RSA server certificate is a CA certificate (BasicConstraints: CA == TRUE !?)
```

4

3.3. MariaDB

- Creación de contraseña para usuario root.
- Se agrego la siguiente linea al archivo my.cnf para deshabilitar la capacidad de cargar archivos locales. Esto puede tener graves implicaciones de seguridad y debe cerrarse a menos que se necesite absolutamente. Esto desactivará la carga de archivos del sistema de archivos para usuarios sin privilegios de nivel de archivo a la base de datos.

```
local-infile=0
```

- Se agregó la siguiente línea al archivo my.cnf para deshabilitar los *symbolic-links*

```
symbolic-links=0
```

- Se verificó que los usuarios existentes para cada host cuenten con una contraseña con el siguiente comando.

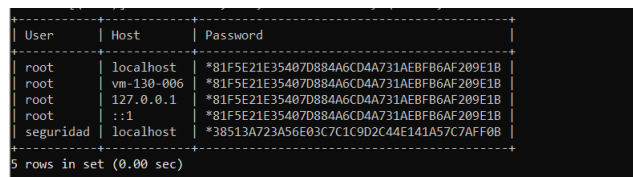
```
SELECT User,Host,Password FROM mysql.user
```

- Se estableció una contraseña para cada usuario existente que no contaba con una en cada host relacionado a la base de datos. Esto se logró por medio del siguiente comando.

```
UPDATE mysql.user SET Password=PASSWORD( 'new_password' )  
WHERE User="username" AND Host='host_name' ;
```

- Se eliminó cada usuario en blanco para la base de datos. Esto se logró por medio del siguiente comando.

```
DELETE FROM mysql.user WHERE User="" ;
```



User	Host	Password
root	localhost	*81F5E21E35407D884A6CD4A731AEBF86AF209E1B
root	vm-130-006	*81F5E21E35407D884A6CD4A731AEBF86AF209E1B
root	127.0.0.1	*81F5E21E35407D884A6CD4A731AEBF86AF209E1B
root	::1	*81F5E21E35407D884A6CD4A731AEBF86AF209E1B
seguridad	localhost	*38513A723A56E03C7C1C9D2C44E141A57C7AFF0B

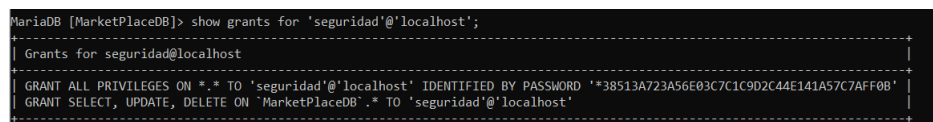
5 rows in set (0.00 sec)

Figura 4: Usuarios de la base de datos.

- Se otorgaron los permisos pertinentes al usuario de la base de datos que pertenece a la aplicación. Esto se logró por medio de los siguientes comandos.

```
FLUSH PRIVILEGES;
```

```
GRANT SELECT,UPDATE,DELETE ON database_name.* TO  
'user'@'localhost' ;
```



```
MariaDB [MarketPlaceDB]> show grants for 'seguridad'@'localhost';
```

Grants for seguridad@localhost
GRANT ALL PRIVILEGES ON *.* TO 'seguridad'@'localhost' IDENTIFIED BY PASSWORD '*38513A723A56E03C7C1C9D2C44E141A57C7AFF0B'
GRANT SELECT, UPDATE, DELETE ON 'MarketPlaceDB'.* TO 'seguridad'@'localhost'

Figura 5: Permisos para el usuario de la base de datos.

- Se cambió el nombre raíz de usuario de inicio de sesión de esa manera si hay un atacante deberá adivinar el nuevo nombre para el host especificado. Esto se realizó con el siguiente comando.

```
rename user 'root'@'localhost' to '4dmin'@'localhost' ;
```

```
MariaDB [MarketPlaceDB]> select user,host,password from mysql.user;
```

user	host	password
admin	localhost	*81F5E21E35407D884A6CD4A731AEBFB6AF209E1B
root	vm-130-006	*81F5E21E35407D884A6CD4A731AEBFB6AF209E1B
root	127.0.0.1	*81F5E21E35407D884A6CD4A731AEBFB6AF209E1B
root	:::1	*81F5E21E35407D884A6CD4A731AEBFB6AF209E1B
seguridad	localhost	*38513A723A56E03C7C1C9D2C44E141A57C7AFF0B

Figura 6: Permisos para el usuario de la base de datos.

3.4. Modulo Home

3.4.1. Objetivos de la seguridad

- Validar la autenticidad de un usuario registrado si es el caso para mostrar los items correctos en la página.

3.4.2. Interacciones

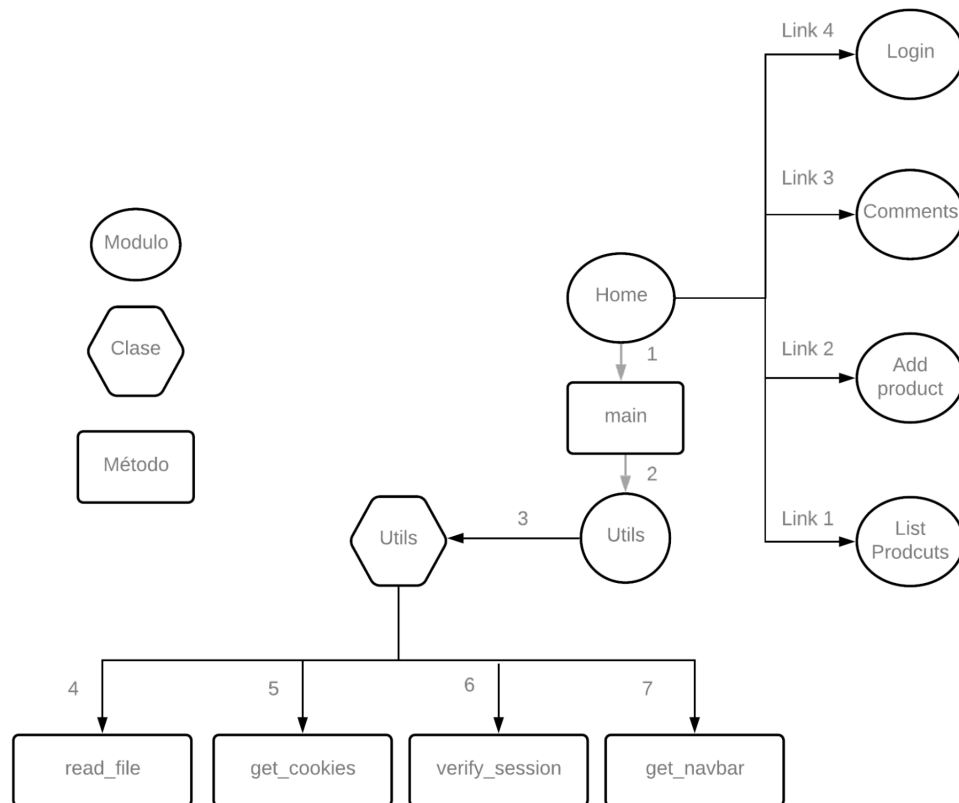


Figura 7: Diagrama de interacción entre todo-y-las-partes módulo Home.

3.4.3. Vulnerabilidades y amenazas

Componentes	Interacción	Servicio(s)	Vulnerabilidad	Amenaza
Login→Utils→Home	5-6-7 Obtención y uso de los cookies para verificar a un usuario.	Integridad, Autenticación	Cookies	Un usuario malintencionado con conocimientos suficientes podría setear cookies para así hacerse pasar por otro usuario y hacer uso de la aplicación.

3.4.4. Riesgos

Componentes	Servicio(s)	Riesgo	Impacto
Login→Utils	Integridad, Confidencialidad	Robo de cookies, implantación de cookies para engañar al sistema.	Alto si se logra ejecutar un script de lectura/implantación de cookies.

3.5. Módulo de Login

3.5.1. Objetivos de la seguridad

- Velar por la **confidencialidad** de la información de inicio de sesión.
- Procurar la **integridad** de la información ingresada por el usuario y la información del sistema.
- **Autenticar** usuarios correctamente.

3.5.2. Interacciones

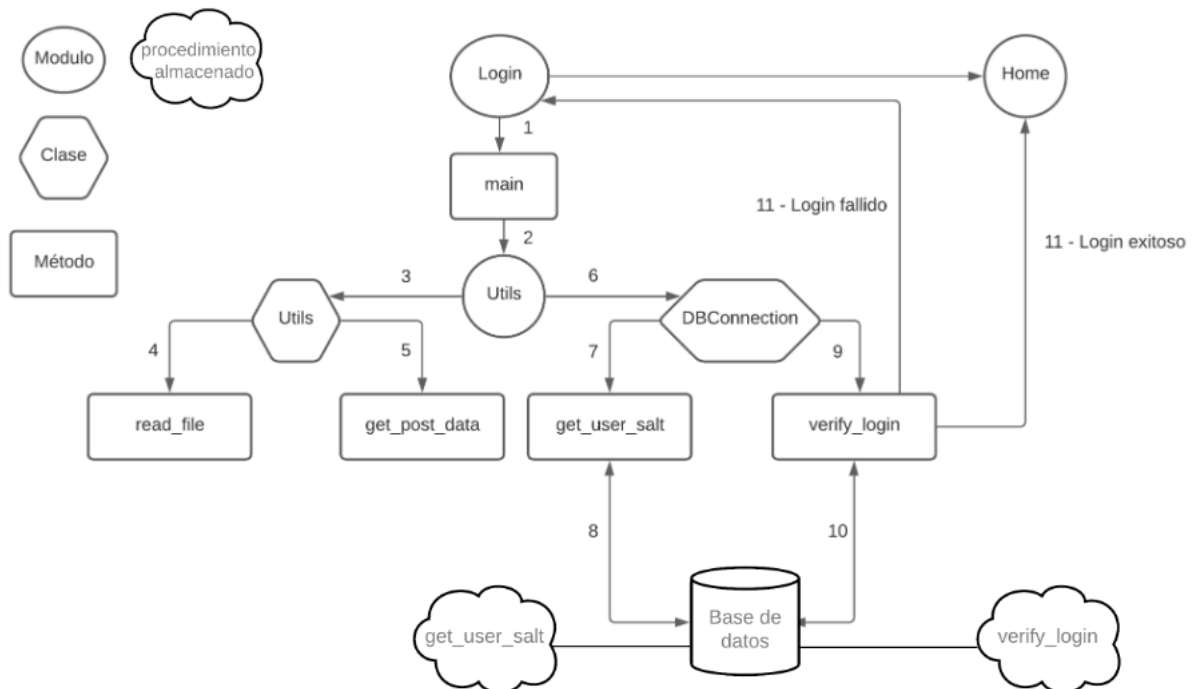


Figura 8: Diagrama de interacción entre todo-y-las-partes módulo Login

3.5.3. Vulnerabilidades y amenazas

Componentes	Interacción	Servicio(s)	Vulnerabilidad	Amenaza
Login→Utils→Home	5-6-7-8-9-10-11 Obtención y uso de los datos del form para obtener la llave de encriptación de la contraseña del usuario y verificar la cuenta.	Integridad, Confidencialidad, Autenticación	El form que recibe entrada de datos por parte del usuario.	Un usuario malintencionado con conocimientos suficientes envía en el form scripts o comandos que puedan comprometer la integridad y confidencialidad de la información al llenar el form de login. Además puede causar buffer overflow si no se controla la entrada de datos.
Login→Utils	5-6-7-8-9-10-11 Múltiples intentos de inicio de sesión.	Autenticación, Integridad	El form que recibe entrada de datos por parte del usuario.	Un usuario malintencionado puede utilizar herramientas automatizadas para forzar el ingreso a la aplicación.

3.5.4. Riesgos

Componentes	Servicio(s)	Riesgo	Impacto
Login→Utils	Integridad	Ataques de Cross Site Scripting	Medio si afecta el funcionamiento de la aplicación. Alto si se realiza un script de robo de información o ejecución de malware.
Login→Utils	Integridad y Confidencialidad	Inyección SQL	Alto si se logra ejecutar una sentencia en la base de datos, como un borrado por ejemplo.
Login→Utils	Autenticidad	Ataques de password cracking.	Alto ya que podría hacerse uso de la aplicación haciéndose pasar por un usuario si se logra hackear la contraseña.
Login→Utils	Integridad y Disponibilidad	Buffer Overflow	Alto si se logra tirar la página o insertar código malicioso.

3.6. Módulo de Registro

3.6.1. Objetivos de la seguridad

- Procurar la **confidencialidad** de la información que ingresa un usuario al registrarse.
- Procurar la **integridad** de la información ingresada por el usuario y la información del sistema al registrar un usuario.

3.6.2. Interacciones

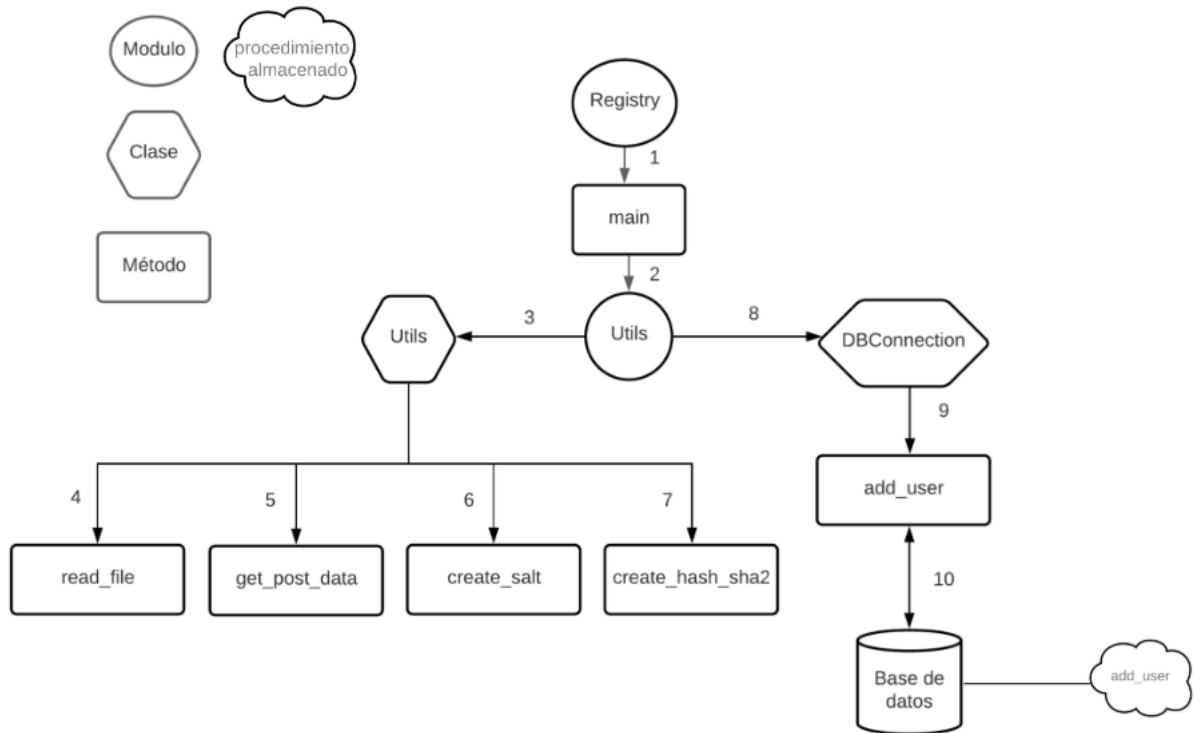


Figura 9: Diagrama de interacción entre todo-y-las-partes módulo Registry

3.6.3. Vulnerabilidades y amenazas

Componentes	Interacción	Servicio(s)	Vulnerabilidad	Amenaza
Registry→Utils	5-6-7-8-9-10 Obtención y uso de los datos del form para almacenar la información del usuario en la base de datos.	Integridad y Confidencialidad	El form que recibe entrada de datos por parte del usuario.	Un usuario malintencionado con conocimientos suficientes envía en el form scripts o comandos que puedan comprometer la integridad y confidencialidad de la información al llenar el form de registro. Además puede causar buffer overflow si no se controla la entrada de datos.
Registry→Utils	5-6-7-8-9-10 Obtención y uso de los datos del form para almacenar la información del usuario y crear una cuenta en la aplicación.	Integridad y Responsabilidad	El form que permite la creación de un usuario.	Un usuario malintencionado con la intención de crear una cuenta con un correo ajeno.

3.6.4. Riesgos

Componentes	Servicio(s)	Riesgo	Impacto
Registry→Utils	Integridad	Ataques de Cross Site Scripting	Medio si afecta el funcionamiento de la aplicación. Alto si se realiza un script de robo de información o ejecución de malware.
Registry→Utils	Integridad y Confidencialidad	Inyección SQL	Alto si se logra ejecutar una sentencia en la base de datos, como un borrado por ejemplo.
Registry→Utils	Integridad y Disponibilidad	Buffer Overflow	Alto si se logra tirar la página o insertar código malicioso.
Registry→Utils	Responsabilidad e Integridad	Suplantación de identidad	Alto en caso de que se cree una cuenta con un correo que no le pertenece a una persona, ya que podría hacerse pasar por alguien más.

3.7. Módulo de Comentarios

3.7.1. Objetivos de la seguridad

- Procurar la **integridad** de la información ingresada por el usuario y la información del sistema.

3.7.2. Interacciones

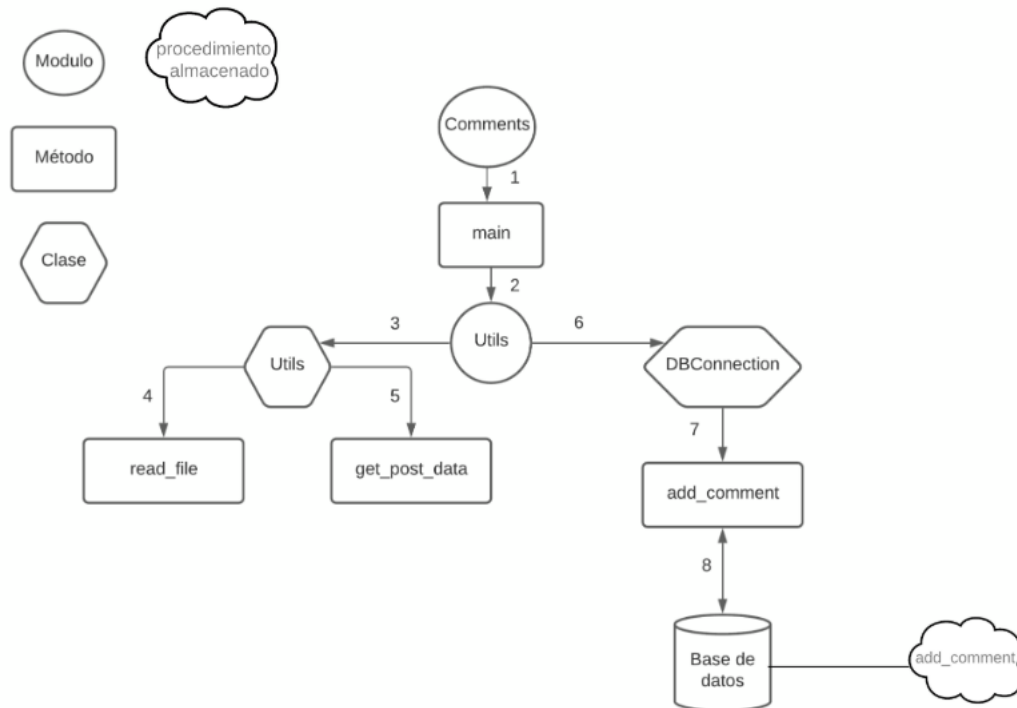


Figura 10: Diagrama de interacción entre todo-y-las-partes módulo Comments

3.7.3. Vulnerabilidades y amenazas

Componentes	Interacción	Servicio(s)	Vulnerabilidad	Amenaza
Comments→Utils	5-6-7-8 Obtención y uso de los datos del form para insertar un nuevo comentario, reclamo o retroalimentación en la base de datos.	Integridad	El form que recibe entrada de datos por parte del usuario.	Un usuario malintencionado con conocimientos suficientes envía en el form scripts o comandos que puedan resultar peligrosos.

3.7.4. Riesgos

Componentes	Servicio(s)	Riesgo	Impacto
Comments→Utils	Integridad	Ataques de Cross Site Scripting	Medio si afecta el funcionamiento de la aplicación. Alto si se realiza un script de robo de información o ejecución de malware.
Comments→Utils	Integridad y Confidencialidad	Inyección SQL	Alto si se logra ejecutar una sentencia en la base de datos, como un borrado por ejemplo.
Comments→Utils	Integridad y Disponibilidad	Buffer Overflow	Alto si se logra tirar la página o insertar código malicioso.
Comments→Utils	Integridad, Confidencialidad	El robo de cookies así como implantación de cookies para engañar al sistema.	Alto si se logra ejecutar un script de lectura/implantación de cookies.

3.8. Módulo para agregar productos

3.8.1. Objetivos de la seguridad

- Procurar la **integridad** de la información ingresada por el usuario en el formulario de agregar producto.
- Procurar la **integridad** de la información almacenada en el sistema que tiene interacción con agregar un producto.
- Procurar la **integridad** del precio.
- Requerir la **autenticación y autorización** de los usuarios para agregar un producto a la venta.

3.8.2. Interacciones

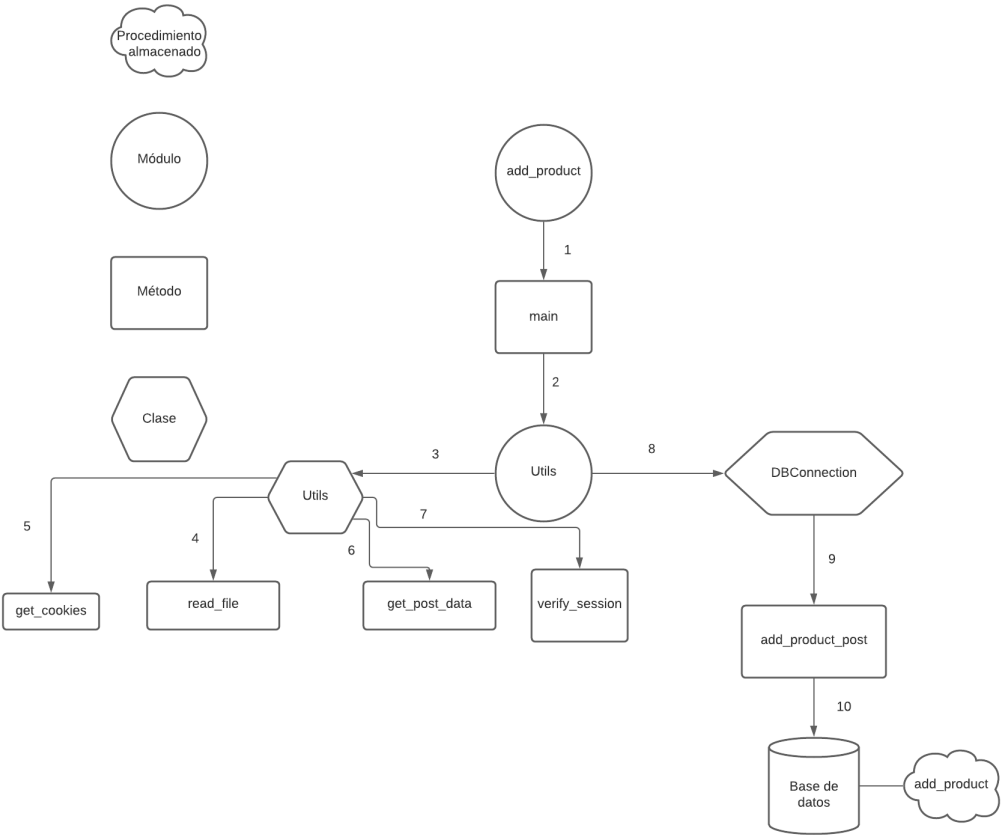


Figura 11: Diagrama de interacción entre todo-y-las-partes módulo Add product

3.8.3. Vulnerabilidades y amenazas

Componentes	Interacción	Servicio(s)	Vulnerabilidad	Amenaza
Add_Product→Utils	4-5-6-7-8 Obtención y uso de los datos del form para agregar un producto para vender.	Integridad	El form que recibe entrada de datos por parte del usuario.	Un usuario malintencionado con conocimientos suficientes envía en el form scripts o comandos que puedan resultar peligrosos.

3.8.4. Riesgos

Componentes	Servicio(s)	Riesgo	Impacto
Add_Product→Utils	Integridad	Ataques de Cross Site Scripting	Medio si afecta el funcionamiento de la aplicación. Alto si se realiza un script de robo de información o ejecución de malware.
Add_Product→Utils	Integridad y Confidencialidad	Inyección SQL	Alto si se logra ejecutar una sentencia en la base de datos, como un borrado por ejemplo.
Add_Product→Utils	Integridad y Disponibilidad	Buffer Overflow	Alto si se logra tirar la página o insertar código malicioso.
Add_Product→Utils	Integridad	Desbordamiento de números	Alto si se ingresa un número muy alto como precio que pueda resultar en desbordar el número que representa el precio. De la misma forma si la suma final del monto total a pagar desbordada.
List products→Utils	Integridad, Confidencialidad	El robo de cookies así como implantación de cookies para engañar al sistema.	Alto si se logra ejecutar un script de lectura/implantación de cookies.

3.9. Módulo para listar productos

3.9.1. Objetivos de la seguridad

- Procurar la **integridad** de la información cuando se buscan productos.
- Procurar la **integridad** de la información cuando se agrega un producto del carrito.
- Procurar la **integridad** de la información del sistema almacenada.
- Requerir la **autenticación** para que solo usuarios registrados puedan agregar productos del carrito.

3.9.2. Interacciones

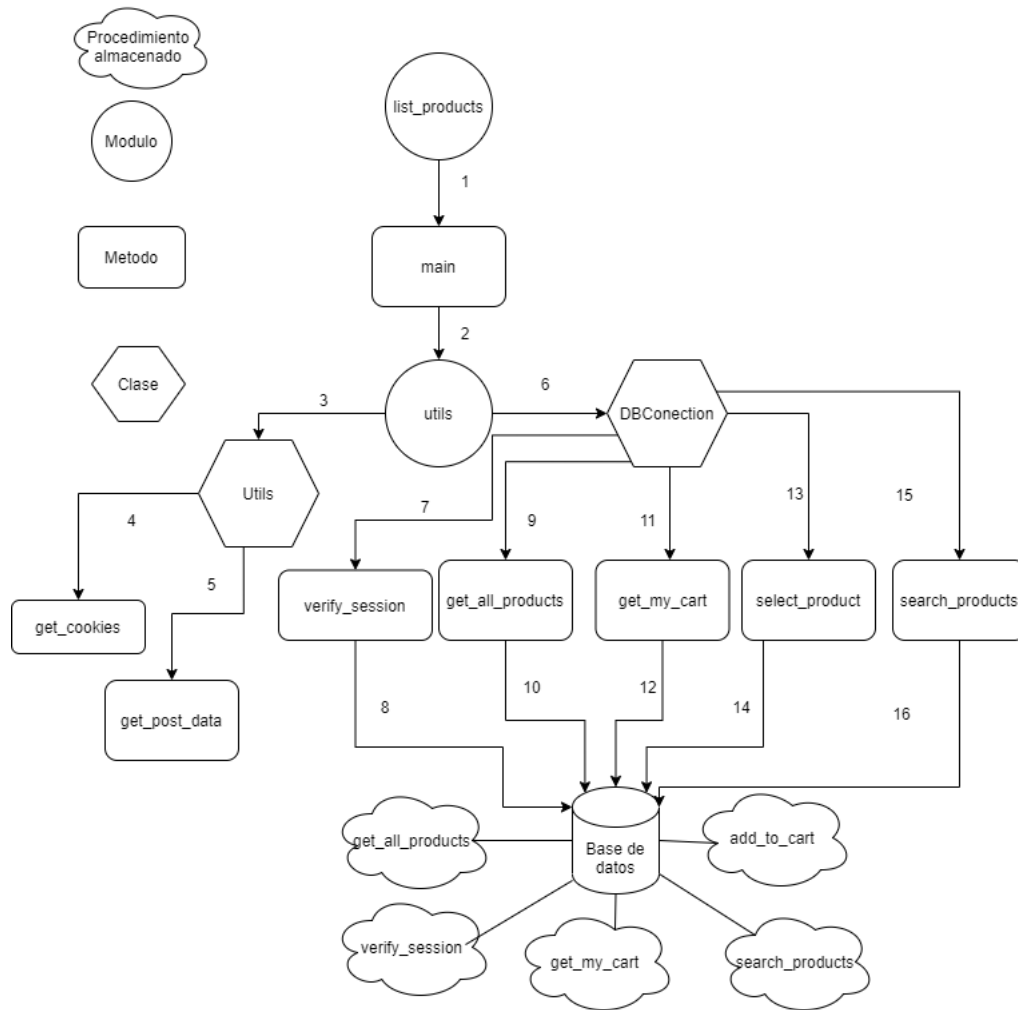


Figura 12: Diagrama de interacción entre todo-y-las-partes módulo List Products

3.9.3. Vulnerabilidades y amenazas

Componentes	Interacción	Servicio(s)	Vulnerabilidad	Amenaza
List products→Utils	6-15-16 Obtención de los datos del form para presentar los productos que el usuario busca	Integridad.	El form recibe entrada de datos de los usuarios	Un usuario con conocimientos maliciosos para inyectar via la entrada de datos un script o una inyeccion de SQL.
List products→Utils	3-4-6-7-8-11-12-13-14 Obtención y uso de los cookies para verificar a un usuario.	Integridad, Autenticacion	Cookies	Un usuario malintencionado con conocimientos suficientes podría setear cookies para así hacerse pasar por otro usuario y observar que productos ha agrgado al carrito y cuales no y añadir productos a su carrito.
List products→Utils	6-15-16 Obtención y uso de los datos del form para buscar un producto de la base de datos	Integridad	El form recibe entrada de datos de los usuarios	Alto si se ingresa un texto de caracteres muy largo para buscar un producto que coincida, puede resultar en desbordar la longitud de la variable y tener problemas al realizar la consulta a la base de datos.

3.9.4. Riesgos

Componentes	Servicio(s)	Riesgo	Impacto
List products→Utils	Integridad	Ataques de Cross Site Scripting	Medio si afecta el funcionamiento de la aplicación. Alto con un script que realice de robo de información o ejecución de malware.
List products→Utils	Integridad y Confidencialidad	Inyección SQL	Alto Alguna sentencia que en la base de datos, logre un borrado o actualización.
List products→Utils	Integridad y Disponibilidad	Buffer Overflow	Alto insertar código malicioso en la entrada de datos.
List products→Utils	Integridad, Confidencialidad	El robo de cookies así como implantación de cookies para engañar al sistema.	Alto si se logra ejecutar un script de lectura/implantación de cookies.

3.10. Módulo para ver un carrito

3.10.1. Objetivos de la seguridad

- Procurar la **integridad** de la información cuando se quita un producto del carrito.
- Procurar la **integridad** de la información cuando se compran los productos del carrito.
- Procurar la **integridad** de la información almacenada en el sistema .
- Rerquerir la **autenticación** para que solo el usuario propietario del carrito pueda eliminar productos del carrito.
- Rerquerir la **autenticación** para que solo el usuario del propietario del carrito pueda visualizar los productos del carrito.
- Rerquerir la **autenticación** para que solo el usuario del propietario del carrito pueda comprar los productos del carrito.
- Procurar la **integridad** de la información de pago ingresada a la hora de realizar una compra.
- Procurar la **autenticidad** de la tarjeta ingresada para realizar una compra.

3.10.2. Interacciones

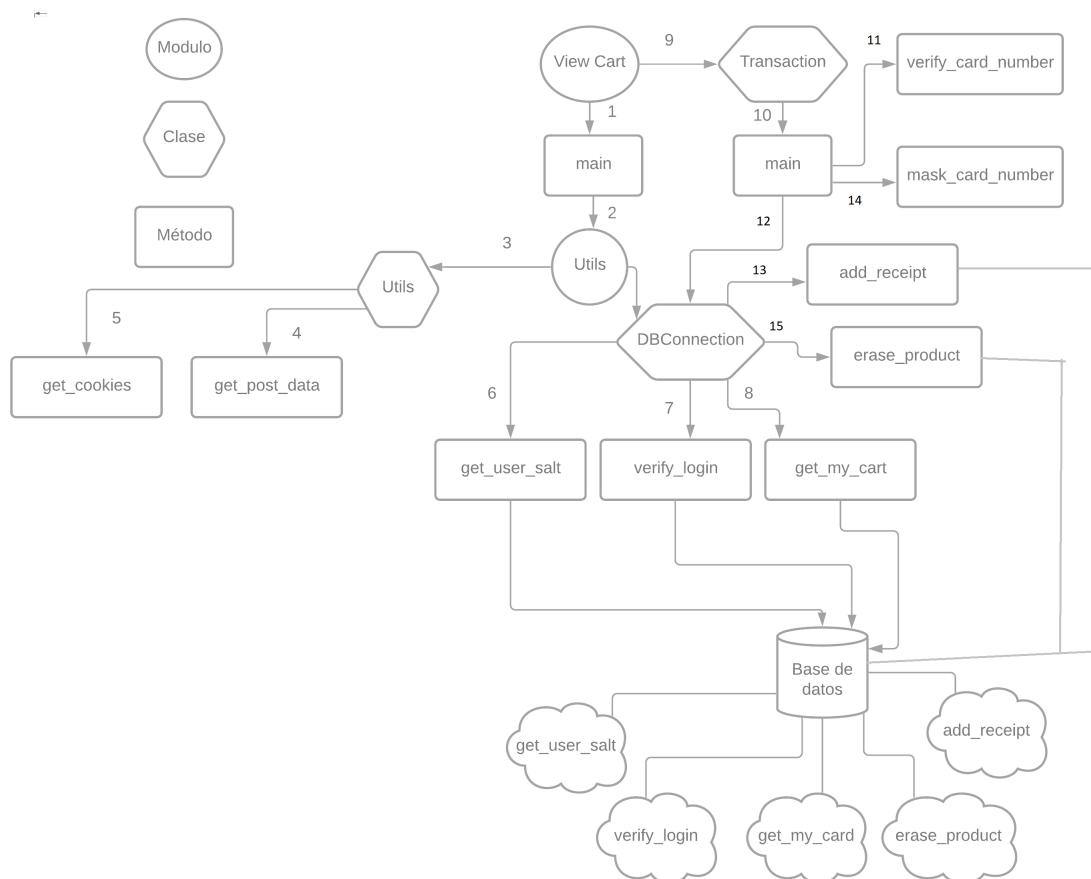


Figura 13: Diagrama de interacción entre todo-y-las-partes módulo View Car

3.10.3. Vulnerabilidades y amenazas

Componentes	Interacción	Servicio(s)	Vulnerabilidad	Amenaza
View Cart→Utils	3-4-5-6-7-8-11-12-13-14-15 Obtención y uso de los cookies para verificar a un usuario.	Integridad, Autenticación	Cookies	Un usuario malintencionado con conocimientos suficientes podría setear cookies para así hacerse pasar por otro usuario, ver los productos del carrito y remover productos de su así como la realización de una compra con ese otro usuario.
View Cart→Utils	3-5 Manejo de los datos obtenidos de la base de datos	Integridad	Desbordamiento de números	Al obtener números muy altos de precios de productos la suma de los precios para calcular el monto total, pueda resultar en desbordamiento de memoria.
View Cart→Transaction	9-10-11-12 Manejo de la información de pago	Integridad - Confidencialidad	Robo de información sensible	Robo de información de una tarjeta de crédito ajena.

3.10.4. Riesgos

Componentes	Servicio(s)	Riesgo	Impacto
View Cart→Utils	Integridad	Desbordamiento de números	Alto si se ingresa un texto alto como precio que pueda resultar en desbordar el número que representa el precio. De la misma forma si la suma final del monto total a pagar desbordada.
View Cart→Utils	Integridad, Confidencialidad	Robo de cookies, implantación de cookies para engañar al sistema.	Alto si se logra ejecutar un script de lectura/implantación de cookies.
View Cart→Transaction	Integridad - Confidencialidad	Robo de información sensible	Alto si se logra robar o tener acceso a la información de una tarjeta de crédito ajena.

4. Políticas y controles de la aplicación web

Componentes	Servicio(s)	Política	Control
Login - Registry - List products - Add products - Comments - View cart	Confidencialidad e Integridad	La información debe ser siempre protegida, cualquiera que sea su forma de ser compartida, comunicada, almacenada o utilizada.	Se utiliza https para encriptar la información que pasa por medio de la red en cada request. Las contraseñas se guardan encriptadas en la base de datos y no en texto plano con el fin de evitar que un atacante pueda adivinarlas, estas son encriptadas en el backend y no en la base de datos. Además, no hay manera de revertir el hash aplicado por lo que para iniciar sesión se necesita de una llave de encriptación para así poder verificar la contraseña ingresada por el usuario.
Login - Registry - Add products - Comments - List Products	Integridad	Toda entrada de datos por parte del usuario debe ser verificada antes de ser utilizada.	Para procesar la información suministrada por los usuarios en los forms se utiliza un método que limpia los datos para eliminar sentencias o comandos que pueden considerarse maliciosos.
Login - Registry - Add products - Comments - List Products - View Cart	Integridad	Toda entrada de datos por parte del usuario debe ser verificada antes de ser utilizada.	Además, se utilizan expresiones regulares y validaciones para los campos de los forms. Se limita la entrada de datos a un número máximo de caracteres según sea el caso para así evitar cadenas de texto excesivamente largas.
Login - Registry - Add products - Comments - List Products - View Cart	Integridad	Toda entrada de datos por parte del usuario debe ser verificada antes de ser utilizada.	Se limita la entrada de datos al tipo pertinente ya sea número o texto para evitar problemas de conversión o uso. De la misma forma, se limita un número mínimo y un número máximo como precio, esto con el objetivo de evitar desbordamiento en las variables que almacenan este atributo, así como en el monto final del precio total (mínimo 1 máximo 10 000 000).
Login - Registry - Add products - Comments - List Products	Integridad	Toda entrada de datos por parte del usuario debe ser verificada antes de ser utilizada.	Para consultas a la base de datos se utilizan consultas parametrizadas o procedimientos almacenados para así evitar inyección SQL.
Login - Registry - List products - Add products - View cart - Comments	Autenticación - Auditoría - Integridad - Disponibilidad	Cada acción en la aplicación debe ser logueada con el fin de contar con un respaldo en caso de errores, problemas o respaldos.	Se implementó un tipo de bitácora donde se registran las acciones de la aplicación ya sean exitosas o erróneas.
List product - Add products - View cart	Autenticación - Disponibilidad - Autorización	Solo usuarios con una sesión iniciada, pueden agregar productos para vender, agregar y quitar productos al carrito y realizar compras.	Se manejan sesiones para cada usuario haciendo uso de cookies para así verificar la identidad del mismo y que este pueda hacer uso de las funciones de la aplicación pertinente. Si un usuario no inicia sesión solo puede consultar los productos existentes.

Componentes	Servicio(s)	Política	Control
View cart	Integridad - Confidencialidad	El uso de tarjetas de crédito debe estar protegido al enviar la información de pago y bajo ninguna circunstancia información sensible debe ser almacenada o compartida.	Se utiliza https para cifrar el contenido de los mensajes enviados vía POST, de esa manera la información de la tarjeta viaja encriptada. Además, para mostrar y almacenar la factura NO se utiliza el cvc y el número de tarjeta no se almacena ni se muestra completo, únicamente se utilizan los 4 últimos dígitos.
View cart	Autorización	Solo tarjetas autorizadas pueden realizar compras.	Se validan las tarjetas de crédito antes de realizar cualquier transacción aplicando el algoritmo de Luhn y además se aplican expresiones regulares para validar los tipos de tarjeta Visa, Mastercard y American Express.