Universidad de Los Andes

Facultad de Ingeniería

Escuela de Sistemas

Mérida Estado Mérida

**MakariónWeb**

**Documentación – Ingeniería de software / Base de Datos**

**Integrantes:**

- Génesis Uzcategui

C.I.: 23.721.718

- José Reinoza

C.I.: 21.332.178

**Nombre del proyecto:** MakariónWeb

**Justificación del proyecto:**

La importancia y necesidad de apoyarse en las tecnologías de la información y comunicación para administrar los procesos, actividades, información y comercio de las empresas viene creciendo cada día más, esto debido a que la tecnología ha provocado muchas transformaciones y avances, que han generado grandes aportes en el rendimiento de las labores cotidianas de una empresa.

Entre las tecnologías se encuentra las páginas web, que por sus grandes características y la capacidad de convertirse en un medio de comunicación universal, han expandido y revolucionado el mundo del comercio, pues le permite a las empresas hacer mayor publicidad de sus productos y servicios, obteniendo de esta manera nuevos clientes potenciales. De la misma manera, permite realizar ventas virtuales, sin la necesidad de que exista contacto presencial entre los clientes y las empresas.

Por todas las ventajas que aporta la presencia en la web, le planteamos a la empresa Creaciones Makarión involucrarse y apoyarse en este medio, a través de la creación de una página web, donde la empresa pueda ampliar sus ventas mediante el comercio electrónico de manera fácil, rápida y sencilla, al mismo tiempo que expande su forma de promocionarse, obteniendo así beneficios para el crecimiento y desarrollo de la empresa.

**Descripción del cliente:**

El cliente de este proyecto es una pequeña empresa llamada Creaciones Makarión, que se enfoca en la venta y fabricación de una serie de productos como cartucheras, bolsos, porta cosméticos, porta laptops, entre otros. Cuenta con una serie de empleados, entre ellos algunos administradores que se encargan de las gestiones más importantes de la empresa, y a los cuales se les darán privilegios en la aplicación web para su respectiva administración.

**Requisitos – Plantillas Volere:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Identificador: R1** | **Tipo de Requisito: Funcional** | | **Caso de Uso: Registrar usuario** |
| ***Descripción:*** *Registrarse*  El administrador principal se registrará manualmente con su nombre, apellido y correo electrónico al momento de programar el sitio web y se le facilitará la contraseña para poder acceder como administrador.  *Para registrar administradores adicionales, el administrador principal debe otorgarles acceso por medio de los correos electrónicos que le indiquen, registrando también los nombres y apellidos, además de crearles una contraseña provisional (que posteriormente cada administrador puede cambiar si lo desea).*  *A los clientes que ingresen al sitio web se les mostrará la página con vista de tienda, donde pueden registrarse dando clic en el botón “Registrarse” o al momento de efectuar una compra se les solicitará que ingresen sus datos para registrarlos en el sistema.*  *Los datos necesarios para registrar tanto a clientes como a administradores son: nombre, apellido, correo electrónico y contraseña.* | | | |
|
|
|
|
| ***Justificación del Requisito:*** *Es indispensable para poder realizar ciertas tareas en el sitio web dependiendo del tipo de usuario que se trate (administrador o cliente).*  *Para los administradores, poder agregar, eliminar o modificar los productos, entre otras cosas; y para los clientes, poder realizar las compras.* | | | |
|
|
|
|
| ***Fuente*** *(que interesado lo propone)****:*** *Creaciones Makarión* | | ***Unidad en la que se origina:*** *Administración y clientes* | |
|
| ***Criterios de validación:*** *El sitio web debe desplegar la página de registro de administrador la primera vez que ingresa un usuario (y mientras no se halla registrado aún el primer administrador).*  *Debe registrar correctamente los datos ingresados por los usuarios.*  *Si un cliente no registrado intenta realizar una compra, debe desplegar la página de registro para clientes conservando los productos que lleva en el carrito.* | | | |
|
|
| ***Grado de Satisfacción: 5*** | | ***Grado de insatisfacción: 5*** | |
| ***Dependencias:*** | | ***Conflictos:*** | |
| ***Documentos de soporte:*** | | ***Histórico de cambios:*** | |
|
| ***Proyecto:*** MakariónWeb | |
| ***Analista:*** *José Reinoza* | | ***Prioridad:*** *5* | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Identificador: R2** | **Tipo de Requisito: Funcional** | | **Caso de Uso: Iniciar sesión** |
| ***Descripción:*** *Iniciar sesión  Todo usuario que previamente se haya registrado en la página, puede iniciar sesión introduciendo su correo electrónico y su respectiva contraseña. Ingresados dichos datos correctamente y seleccionando la opción ingresar, se desplegará el portal y las operaciones correspondientes al modo de usuario que ingreso: usuario cliente/administrador.* | | | |
|
|
|
|
| ***Justificación del Requisito:*** *Es necesario poder iniciar sesión he ingresar a la página como un usuario autenticado, ya que ciertas operaciones que provee la página, cuentan con restricciones que solo se desbloquean para ciertos usuarios de la página y si se encuentran registrados y autenticados, por ejemplo: como cliente hacer una compra, o como administrador eliminar un determinado producto.* | | | |
|
|
|
|
| ***Fuente*** *(que interesado lo propone)****:*** *Creaciones Makarión* | | ***Unidad en la que se origina:*** *Administración y clientes* | |
|
| ***Criterios de validación:*** *La página debe proveer dos campos para ingresar los datos necesarios para poder iniciar sesión. Ingresado los datos, la página abre correctamente para cada usuario.* | | | |
|
|
| ***Grado de Satisfacción: 5*** | | ***Grado de insatisfacción: 5*** | |
| ***Dependencias:*** *R1* | | ***Conflictos:*** | |
| ***Documentos de soporte:*** | | ***Histórico de cambios:*** | |
|
| ***Proyecto:*** MakariónWeb | |
| ***Analista:*** *Génesis Uzcategui* | | ***Prioridad:*** *5* | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Identificador: R3** | **Tipo de Requisito: Funcional** | | **Caso de Uso: Registrar producto** |
| ***Descripción:*** *Registrar producto Registra los productos que se desean vender a través de la página web, con los siguientes datos:*  - Código del producto  - Modelo  - Precio  - Color  - Categoría: Nombre, características.  Una vez ingresados se guardan en la base de datos de la página. | | | |
|
|
|
|
| ***Justificación del Requisito:*** *Es necesario ingresar el producto en la página, para poder mostrar y promocionar el producto a los futuros visitantes de la página, así como también poder observarlo como administrador y ver que se encuentra registrado correctamente.* | | | |
|
|
|
|
| ***Fuente*** *(que interesado lo propone)****:*** *Creaciones Makarión* | | ***Unidad en la que se origina:*** *Administración* | |
|
| ***Criterios de validación:*** *La página debe proveer una serie de campos para el ingreso de los datos necesarios a registrar del producto, así como también la opción a guardar. Se registran correctamente los datos ingresados.* | | | |
|
|
| ***Grado de Satisfacción: 5*** | | ***Grado de insatisfacción: 5*** | |
| ***Dependencias:*** *R1 y R2* | | ***Conflictos:*** | |
| ***Documentos de soporte:*** | | ***Histórico de cambios:*** | |
|
| ***Proyecto:*** MakariónWeb | |
| ***Analista:*** *José Reinoza* | | ***Prioridad:*** *5* | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Identificador: R4** | **Tipo de Requisito: Funcional** | | **Caso de Uso: Buscar producto** |
| ***Descripción:*** *Consultar producto El sistema les permite a los administradores buscar un producto en la base de datos, ya sea por medio de su código de identificación, o ingresando su respectivo nombre. El sistema muestra por pantalla datos básicos como: el precio, código, entre otros.* | | | |
|
|
|
|
| ***Justificación del Requisito:*** *Es necesario para poder encontrar los productos en el sistema y ver la información pertinente sobre los mismos, así como también para poder modificar un determinado dato o eliminar dicho producto.* | | | |
|
|
|
|
| ***Fuente*** *(que interesado lo propone)****:*** *Creaciones Makarión* | | ***Unidad en la que se origina:*** *Administración* | |
|
| ***Criterios de validación:*** *- Todos los datos mostrados deben corresponderse con los especificados para el producto.* | | | |
|
|
| ***Grado de Satisfacción: 4*** | | ***Grado de insatisfacción: 5*** | |
| ***Dependencias:*** *R1, R2 y R3* | | ***Conflictos:*** | |
| ***Documentos de soporte:*** | | ***Histórico de cambios:*** | |
|
| ***Proyecto:***MakariónWeb | |
| ***Analista:*** *José Reinoza* | | ***Prioridad:*** *4* | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Identificador: R5** | **Tipo de Requisito: Funcional** | | **Caso de Uso: Modificar producto** |
| ***Descripción:*** *Modificar producto Modificar producto es una opción que aparece en la página una vez que se ha realizado una búsqueda y seleccionado un producto en específico, desplegándose todos los datos referentes a dicho producto. Esta opción, permite actualizar cualquiera de los campos de información que se encuentran predefinidos para los productos como: código, modelo, entre otros.*  *Al terminar de actualizar los datos e información que se desea, modificar producto provee la opción guardar, para que los cambios realizados, queden registrados en la base de datos de la página, y estos se puedan observar al momento de realizar nuevamente esta búsqueda.* | | | |
|
|
|
|
| ***Justificación del Requisito:*** *Es indispensable poder modificar los datos de un producto, ya que estos datos pueden variar en algunas ocasiones, como por ejemplo el incremento o la rebaja del precio de un producto específico, también puede ocurrir que algún dato registrado anteriormente se halla colocado incorrecto, por lo que se desee corregir.* | | | |
|
|
|
|
| ***Fuente*** *(que interesado lo propone)****:*** *Creaciones Makarión* | | ***Unidad en la que se origina:*** *Administración* | |
|
| ***Criterios de validación:*** *- Despliegue de los campos predefinidos para poder editar los datos del producto.*  *- Al buscar nuevamente el producto, muestra los datos correctamente modificados.* | | | |
|
|
| ***Grado de Satisfacción: 3*** | | ***Grado de insatisfacción: 3*** | |
| ***Dependencias:*** *R1, R2, R3 y R4* | | ***Conflictos:*** | |
| ***Documentos de soporte:*** | | ***Histórico de cambios:*** | |
|
| ***Proyecto:***MakariónWeb | |
| ***Analista:***Génesis Uzcategui | | ***Prioridad:*** *3* | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Identificador: R6** | **Tipo de Requisito: Funcional** | | **Caso de Uso: Eliminar producto** |
| ***Descripción:*** *Eliminar producto Eliminar producto es una opción que se despliega una vez que se ha realizado una búsqueda y seleccionado un producto en específico. Permite eliminar todos los datos referentes a esta búsqueda de manera definitiva de la base de datos del sistema donde se encuentran almacenados.* | | | |
|
|
|
|
| ***Justificación del Requisito:*** *Es necesario este requisito, para poder actualizar la base de datos, eliminando productos que ya no se fabriquen en la empresa o que el producto tenga algún error y se desee eliminar por completo sus respectivos datos, dejando en la base de datos solo los productos que se quieren mostrar en la página web.* | | | |
|
|
|
|
| ***Fuente*** *(que interesado lo propone)****:*** *Creaciones Makarión* | | ***Unidad en la que se origina:*** *Administración* | |
|
| ***Criterios de validación:*** *- Al buscar el producto después de la eliminación la página muestra un aviso indicando que el producto no se encuentra en la base de datos.* | | | |
|
|
| ***Grado de Satisfacción: 2*** | | ***Grado de insatisfacción: 3*** | |
| ***Dependencias:*** *R1, R2, R3 y R4* | | ***Conflictos:*** | |
| ***Documentos de soporte:*** | | ***Histórico de cambios:*** | |
|
| ***Proyecto:***MakariónWeb | |
| ***Analista:***Génesis Uzcategui | | ***Prioridad:*** *3* | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Identificador: R7** | **Tipo de Requisito: Funcional** | | **Caso de Uso: Buscar cliente** |
| ***Descripción:*** *Buscar/Ver cliente El sistema le permite a los administradores buscar en la base de datos toda la información registrada sobre un cliente en específico, ingresando como criterio de búsqueda su número de cédula.* | | | |
|
|
|
|
| ***Justificación del Requisito:*** *Necesario para poder ver la información suministrada por el cliente, que sirva para contactarlo en caso de que la empresa lo requiera. Al mismo tiempo es útil para la empresa poder ver datos ingresados, como la edad, pues permite hacer estudios estadísticos y ver por ejemplo, el promedio de edades que utilizan la página, entre otras cosas.* | | | |
|
|
|
|
| ***Fuente*** *(que interesado lo propone): Creaciones Makarión* | | ***Unidad en la que se origina:*** *Administración* | |
|
| ***Criterios de validación:*** *- La información proporcionada por la base de datos muestra todos los atributos del cliente* | | | |
|
|
| ***Grado de Satisfacción: 4*** | | ***Grado de insatisfacción: 5*** | |
| ***Dependencias:*** *R1 y R2* | | ***Conflictos:*** | |
| ***Documentos de soporte:*** | | ***Histórico de cambios:*** | |
| ***Proyecto:***MakariónWeb | |
| ***Analista:*** *José Reinoza* | | ***Prioridad:*** *2* | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Identificador: R8** | **Tipo de Requisito: Funcional** | | **Caso de Uso: Buscar cliente** |
| ***Descripción:*** *Eliminar cliente El sistema busca en la base de datos al cliente indicado por su número de cédula y elimina toda la información asociada a él.* | | | |
|
|
|
|
| ***Justificación del Requisito:*** *Necesario para eliminar la información asosiada a un clientesuministrada por el cliente, que sirva para contactarlo en caso de que la empresa lo requiera. Al mismo tiempo es útil para la empresa poder ver datos ingresados, como la edad, pues permite hacer estudios estadísticos y ver, por ejemplo, el promedio de edades que utilizan la página, entre otras cosas.* | | | |
|
|
|
|
| ***Fuente*** *(que interesado lo propone): Creaciones Makarión* | | ***Unidad en la que se origina:*** *Administración* | |
|
| ***Criterios de validación:*** *- La información proporcionada por la base de datos muestra todos los atributos del cliente* | | | |
|
|
| ***Grado de Satisfacción: 4*** | | ***Grado de insatisfacción: 5*** | |
| ***Dependencias:*** *R1, R2 y R7* | | ***Conflictos:*** | |
| ***Documentos de soporte:*** | | ***Histórico de cambios:*** | |
|
| ***Proyecto:***MakariónWeb | |
| ***Analista:*** *José Reinoza* | | ***Prioridad:*** *1* | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Identificador: R9** | **Tipo de Requisito: Funcional** | | **Caso de Uso: Buscar venta** |
| ***Descripción:*** *Buscar/Ver compras La página le permite a los administradores buscar en la base de datos una determinada compra realizada por el cliente, por medio de su número de compra. Una vez ingresado este dato, se muestra en pantalla todos los datos almacenados, referentes a dicha compra.* | | | |
|
|
|
|
| ***Justificación del Requisito:*** *Necesario para poder identificar una compra realizada por un cliente, permitiendo observar datos de gran interés para la empresa, como el precio total de la compra, el cliente que la realiza, entre otros datos que permiten llevar un control de los productos que se venden por la página, así como también poder realizar futuras entregas de los productos.* | | | |
|
|
|
|
| ***Fuente*** *(que interesado lo propone): Creaciones Makarión* | | ***Unidad en la que se origina:*** *Administración* | |
|
| ***Criterios de validación:*** *- Muestra correctamente los datos de la compra.*  *- Si la compra no existe, arroja un mensaje por pantalla indicando que la compra no se realizó.* | | | |
|
|
| ***Grado de Satisfacción: 3*** | | ***Grado de insatisfacción: 3*** | |
| ***Dependencias:*** *R1, R2 y R15* | | ***Conflictos:*** | |
| ***Documentos de soporte:*** | | ***Histórico de cambios:*** | |
|
| ***Proyecto:***MakariónWeb | |
| ***Analista:***Génesis Uzcategui | | ***Prioridad:*** *4* | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Identificador: R10** | **Tipo de Requisito: Funcional** | | **Caso de Uso: Buscar producto** |
| ***Descripción:*** *Buscar producto La página le permite al usuario buscar productos que se encuentran almacenados en la base de datos, ingresando el nombre o parte de este. El sistema despliega una lista con el o los productos pertenecientes o relacionados al criterio de búsqueda ingresado.* | | | |
|
|
|
|
| ***Justificación del Requisito:*** *Para seleccionar un producto, el usuario debe previamente localizarlo en la base de datos de la página, por lo que la opción buscar es necesaria para poder encontrar los productos y posteriormente ver la información pertinente sobre los mismos.* | | | |
|
|
|
|
| ***Fuente*** *(que interesado lo propone)****:*** *Creaciones Makarión* | | ***Unidad en la que se origina:*** *Administración, cliente y usuario no autenticado* | |
|
| ***Criterios de validación:*** *- Todos los datos mostrados deben corresponderse con los especificados para el producto.* | | | |
|
|
| ***Grado de Satisfacción: 4*** | | ***Grado de insatisfacción: 5*** | |
| ***Dependencias:*** *R1 y R2* | | ***Conflictos:*** | |
| ***Documentos de soporte:*** | | ***Histórico de cambios:*** | |
|
| ***Proyecto:***MakariónWeb | |
| ***Analista:*** *José Reinoza* | | ***Prioridad:*** *4* | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Identificador: R11** | **Tipo de Requisito: Funcional** | | **Caso de Uso: Modificar producto** |
| ***Descripción:*** *Ver producto (cliente)* La página provee la opción ver producto, con la cual, el cliente registrado o no, puede ver información más detallada acerca del producto que desea. Seleccionando dicho producto, el sistema muestra en la pantalla más detalles de este, que se encuentran almacenados en la base de datos. | | | |
|
|
|
|
| ***Justificación del Requisito:*** *Es indispensable para que el cliente detalle todas las características del producto y de esta manera decida sobre la realización de la compra o no de dicho producto en base a la información que allí se muestra. También es importante la opción ver producto, porque al seleccionarla, se despliega añadir al carrito, opción que es imprescindible para poder comprar el producto.* | | | |
|
|
|
|
| ***Fuente*** *(que interesado lo propone)****:*** *Creaciones Makarión* | | ***Unidad en la que se origina:*** *Administración, cliente y usuario no autenticado* | |
|
| ***Criterios de validación:*** *El sistema debe proveer una opción para la selección de productos en base a cada uno de los criterios de búsqueda ingresados.* | | | |
|
|
| ***Grado de Satisfacción: 3*** | | ***Grado de insatisfacción: 3*** | |
| ***Dependencias:*** *R1, R2 y R3* | | ***Conflictos:*** | |
| ***Documentos de soporte:*** | | ***Histórico de cambios:*** | |
|
| ***Proyecto:***MakariónWeb | |
| ***Analista:***Génesis Uzcategui | | ***Prioridad:*** *4* | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Identificador: R12** | **Tipo de Requisito: Funcional** | | **Caso de Uso:** |
| ***Descripción:*** *Añadir al carrito*  La página provee la opción añadir a carrito, la cual muestra la cantidad, total y tipo de los productos que se han seleccionado para una futura compra. Esta opción puede ser utilizada por clientes registrados o no en la página. Al seleccionar uno de los productos que se encuentra registrados en la base de datos, este se añade desde la pantalla de los detalles del producto seleccionando la opción: Añadir a carrito. | | | |
|
|
|
|
| ***Justificación del Requisito:*** *Es indispensable poder añadir productos al carrito, ya que por medio de esta opción, los clientes registrados y autenticados pueden optar por comprar los productos que aquí se encuentran seleccionados.* | | | |
|
|
|
|
| ***Fuente*** *(que interesado lo propone): Creaciones Makarión* | | ***Unidad en la que se origina:*** *Usuario no autenticado y cliente* | |
|
| ***Criterios de validación:*** *El sistema debe proveer los detalles del producto, con su respectivo recuadro que permite añadirlo al carrito.* | | | |
|
|
| ***Grado de Satisfacción: 4*** | | ***Grado de insatisfacción: 5*** | |
| ***Dependencias:*** *R1, R2 y R3* | | ***Conflictos:*** | |
| ***Documentos de soporte:*** | | ***Histórico de cambios:*** | |
|
| ***Proyecto:***MakariónWeb | |
| ***Analista:*** *Génesis Uzcategui* | | ***Prioridad:*** *4* | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Identificador: R13** | **Tipo de Requisito: Funcional** | | **Caso de Uso:** |
| ***Descripción:*** *Ver carrito*  Se redirige a los clientes a una página donde pueden ver los productos que han agregado al carrito junto con el monto total a pagar y el botón de “Comprar”. Esta opción puede ser utilizada por clientes registrados o no en la página. | | | |
|
|
|
|
| ***Justificación del Requisito:*** *Los clientes lo necesitan para ver el listado de productos que han agregado y la cantidad que desean comprar. Además, les permite posteriormente eliminar cualquier producto que ya no deseen comprar.* | | | |
|
|
|
|
| ***Fuente*** *(que interesado lo propone): Creaciones Makarión* | | ***Unidad en la que se origina:*** *Usuario no autenticado y cliente* | |
|
| ***Criterios de validación:*** *El sistema debe mostrarles a los clientes tanto registrados como no registrados, el listado de productos que han ido agregando al carrito.*  *También debe mostrarse correctamente el monto total a pagar por los productos que ha agregado al carrito.* | | | |
|
|
| ***Grado de Satisfacción: 5*** | | ***Grado de insatisfacción: 4*** | |
| ***Dependencias:*** *R1 y R2* | | ***Conflictos:*** | |
| ***Documentos de soporte:*** | | ***Histórico de cambios:*** | |
|
| ***Proyecto:***MakariónWeb | |
| ***Analista:*** *José Reinoza* | | ***Prioridad:*** *4* | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Identificador: R14** | **Tipo de Requisito: Funcional** | | **Caso de Uso:** |
| ***Descripción:*** *Eliminar del carrito.*  Al ingresar al carrito, la página muestra todos los artículos que allí se encuentran agregados previamente, junto con la opción eliminar disponible para cada producto. El cliente puede borrar todos los artículos que desee, seleccionando la opción eliminar correspondiente al producto que desea borrar. Al ejecutar esta opción, la página se actualiza y deja de mostrar el producto eliminado del carrito. | | | |
|
|
|
|
| ***Justificación del Requisito:*** *Es indispensable para corregir casos donde el cliente agrega al carrito un producto equivocado, o simplemente ya no lo desea comprar, y por ende lo desea eliminar del carrito.* | | | |
|
|
|
|
| ***Fuente*** *(que interesado lo propone): Creaciones Makarión* | | ***Unidad en la que se origina:*** *Usuario no autenticado y cliente* | |
|
| ***Criterios de validación:*** *Se eliminaron correctamente del carrito los productos seleccionados.* | | | |
|
|
| ***Grado de Satisfacción: 4*** | | ***Grado de insatisfacción: 5*** | |
| ***Dependencias:*** *R1, R2, R12 y R13* | | ***Conflictos:*** | |
| ***Documentos de soporte:*** | | ***Histórico de cambios:*** | |
|
| ***Proyecto:***MakariónWeb | |
| ***Analista:*** *Génesis Uzcategui* | | ***Prioridad:*** *3* | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Identificador: R15** | **Tipo de Requisito: Funcional** | | **Caso de Uso:** |
| ***Descripción:*** *Comprar/Pagar*  *Al ingresar al carrito y tener productos agregados a este, se genera la opción pagar. Con esta opción el cliente registrado, puede finalizar su respectiva compra. El sistema actúa guardando los datos de la compra en la base de datos, datos básicos como el número de compra, cliente y fecha, así como también datos más detallados como cantidad, total, entre otros. Al mismo tiempo genera una factura virtual con dichos datos, incorporando información importante para que el cliente pueda pagar, como el número de cuenta de la empresa.* | | | |
|
|
|
|
| ***Justificación del Requisito:***  *Es indispensable para los clientes poder realizar sus compras.*  *Además, también se necesita para registrar las ventas, así esta información queda almacenada en la base de datos del sistema, permitiendo llevar el control de cada venta realizada, que posteriormente se pueda consultar.* | | | |
|
|
|
|
| ***Fuente*** *(que interesado lo propone): Creaciones Makarión* | | ***Unidad en la que se origina:*** *Administración* | |
|
| ***Criterios de validación:*** *- La información de la compra se guarda correctamente. (Apoyarse en Buscar ventas para ver que los datos se guardaron correctamente)* | | | |
|
|
| ***Grado de Satisfacción: 4*** | | ***Grado de insatisfacción: 5*** | |
| ***Dependencias:*** *R1, R2, R12 y R13* | | ***Conflictos:*** | |
| ***Documentos de soporte:*** | | ***Histórico de cambios:*** | |
|
| ***Proyecto:***MakariónWeb | |
| ***Analista:*** *Génesis Uzcategui* | | ***Prioridad:*** *4* | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Identificador: R16** | **Tipo de Requisito: Funcional** | | **Caso de Uso:** |
| ***Descripción:*** *Ver perfil*  *Los clientes pueden ver su información personal dando clic en Perfil en la parte superior de la pantalla. Se muestra su nombre, apellido, correo electrónico y cédula.*  *El campo contraseña se muestra con la contraseña oculta.* | | | |
|
|
|
|
| ***Justificación del Requisito:***  *Los clientes lo necesitan para ver su información registrada en la tienda. Además, sirve para modificar posteriormente sus datos en casos de que lo requieran.* | | | |
|
|
|
|
| ***Fuente*** *(que interesado lo propone): Creaciones Makarión* | | ***Unidad en la que se origina:*** *Administración* | |
|
| ***Criterios de validación:*** *- Carga correctamente la página de perfil con todos los datos al dar clic en “Perfil”.* | | | |
|
|
| ***Grado de Satisfacción: 4*** | | ***Grado de insatisfacción: 4*** | |
| ***Dependencias:*** *R1 y R2* | | ***Conflictos:*** | |
| ***Documentos de soporte:*** | | ***Histórico de cambios:*** | |
|
| ***Proyecto:***MakariónWeb | |
| ***Analista:*** *José Reinoza* | | ***Prioridad:*** *4* | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Identificador: R17** | **Tipo de Requisito: Funcional** | | **Caso de Uso:** |
| ***Descripción:*** *Editar perfil*  *Los clientes pueden editar su información personal registrada en la tienda. Pueden editar su nombre, apellido, correo electrónico, cédula y contraseña.* | | | |
|
|
|
|
| ***Justificación del Requisito:***  *Necesario para corregir posibles errores de los clientes al ingresar su información personal al momento de registrarse* | | | |
|
|
|
|
| ***Fuente*** *(que interesado lo propone): Creaciones Makarión* | | ***Unidad en la que se origina:*** *Administración* | |
|
| ***Criterios de validación:*** *- Se modifican correctamente los datos del cliente en la base de datos.* | | | |
|
|
| ***Grado de Satisfacción: 4*** | | ***Grado de insatisfacción: 4*** | |
| ***Dependencias:*** *R1, R2 y R16* | | ***Conflictos:*** | |
| ***Documentos de soporte:*** | | ***Histórico de cambios:*** | |
|
| ***Proyecto:***MakariónWeb | |
| ***Analista:*** *José Reinoza* | | ***Prioridad:*** *4* | |

**Priorización de requisitos:**

Se utilizará la prioridad indicada en las plantillas volere, siendo 5 la de mayor prioridad.

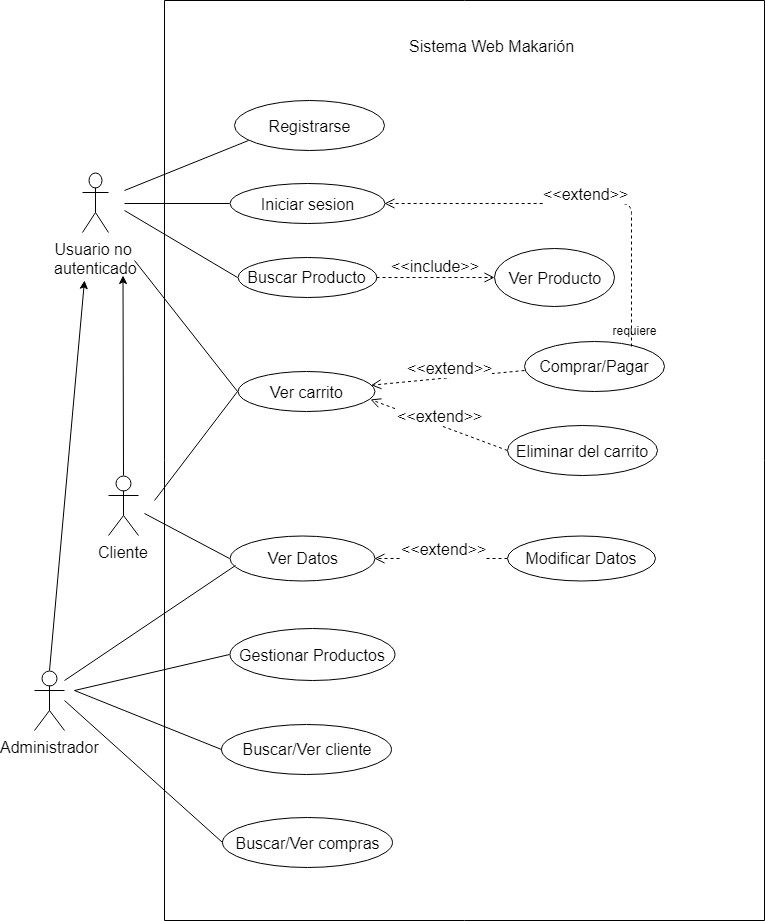
Además, se tomará en cuenta si los requisitos dependen de otros, siendo los más independientes los que se implementarán primero.

**Matriz de dependencias entre requisitos:**

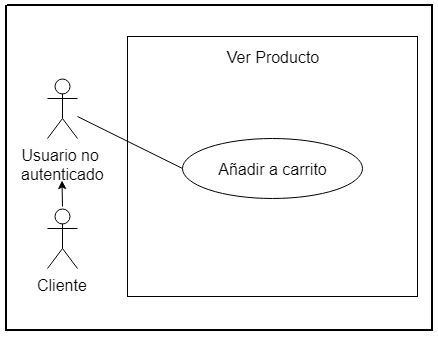
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | RF1 | RF2 | RF3 | RF4 | RF5 | RF6 | RF7 | RF8 | RF9 | RF10 | RF11 | RF12 | RF13 | RF14 | RF15 | RF16 | RF17 |
| RF1 |  | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X |
| RF2 |  |  | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X |
| RF3 |  |  |  | X | X | X |  |  |  |  | X | X |  |  |  |  |  |
| RF4 |  |  |  |  | X | X |  |  |  |  |  |  |  |  |  |  |  |
| RF5 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| RF6 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| RF7 |  |  |  |  |  |  |  | X |  |  |  |  |  |  |  |  |  |
| RF8 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| RF9 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| RF10 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| RF11 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| RF12 |  |  |  |  |  |  |  |  |  |  |  |  |  | X | X |  |  |
| RF13 |  |  |  |  |  |  |  |  |  |  |  |  |  | X | X |  |  |
| RF14 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| RF15 |  |  |  |  |  |  |  |  | X |  |  |  |  |  |  |  |  |
| RF16 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | X |
| RF17 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

**Casos de usos (diagramas y descripción textual):**

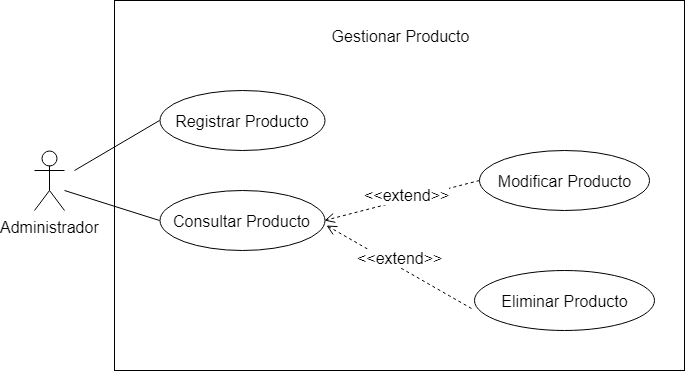
**-** Diagrama de casos de uso general de la Página Web de la empresa Creaciones Makarión.



- Diagrama Caso de Uso específico Ver Producto.



- Diagrama Caso de Uso específico Gestionar Producto.



**Casos de usos (descripción textual):**

**Caso de uso:** Registrarse

**Actores participantes:** Primer administrador, clientes

**Condiciones de entrada:**

* Los clientes pueden acceder a la función registrarse por medio del botón de “Registrarse” que aparece en la parte superior de la tienda. Administradores adicionales deben ser registrados por el administrador principal o por otros administradores.
* También se accede a la función registrarse antes de que un cliente intente realizar una compra sin haber iniciado sesión en la tienda.

**Flujo de eventos:**

*Si se quiere registrar a otro(s) administrador(es):*

1. *El administrador principal u otro adicional que también esté registrado ingresa a su panel de administrador dando clic en el botón “Panel”.*
2. *Se despliega el menú para administradores.*
3. *Da clic en “Agregar administrador”.*
4. *Se despliega una página con un formulario donde debe ingresar el nombre, apellido, cédula, correo electrónico y contraseña para el nuevo administrador.*
5. *El administrador da clic en “Registrar”.*
6. *El sistema verifica que sea único el correo electrónico y que los datos no tengan caracteres inválidos.*
7. *Se guardan los datos del nuevo administrador en la base de datos.*
8. *Se muestra por pantalla el mensaje “Administrador registrado”.*

*Si es un cliente el que se va a registrar antes de hacer una compra:*

1. *El cliente da clic en el botón “Registrar” ubicado en la parte superior de la tienda.*
2. *Se redirige a la página con el formulario de registro.*
3. *El cliente ingresa el nombre, apellido, cédula, correo electrónico y contraseña.*
4. *El cliente da clic en “Registrarme”.*
5. *El sistema verifica que sea único el correo electrónico y que los datos no tengan caracteres inválidos.*
6. *Se guardan los datos del cliente en la base de datos.*
7. *Se muestra por pantalla el mensaje “Cliente registrado”.*
8. *El cliente es redirigido a la página principal de la tienda con su sesión iniciada.*

*Si es un cliente el que se va a registrar en el momento de hacer una compra:*

1. *El cliente da clic en “Comprar”.*
2. *El sistema verifica si el cliente ha iniciado sesión.*
3. *Se le solicita al cliente que se registre si es un usuario nuevo o que ingrese si ya está registrado.*
4. *El cliente elige la opción “Registrarme”.*
5. *Se guardan los productos del carrito de compra del cliente.*
6. *Se despliega el formulario para el registro.*
7. *El cliente ingresa el nombre, apellido, cédula, correo electrónico y contraseña.*
8. *El cliente da clic en “Registrarme”.*
9. *El sistema verifica que sea único el correo electrónico y que los datos no tengan caracteres inválidos.*
10. *Se guardan los datos del cliente en la base de datos.*
11. *Se muestra por pantalla el mensaje “Cliente registrado”.*
12. *Se redirige al cliente a la página de compra con los productos que tenía en el carrito.*

**Condiciones de salida:** Después de mostrar el mensaje “Administrador registrado” o “Cliente registrado” y de ser redirigido a la página principal de la tienda o reanudar la compra pendiente.

**Flujos alternativos:**

6a.1 Si todos los campos no están llenos o hay caracteres inválidos, se vuelve a desplegar la página con el formulario y se marcan en rojo las letras del formulario de los campos inválidos. También se muestra un aviso indicando “Algunos campos son inválidos”.

5b.1 Si todos los campos no están llenos o hay caracteres inválidos, se vuelve a desplegar la página con el formulario y se marcan en rojo las letras del formulario de los campos inválidos. También se muestra un aviso indicando “Algunos campos son inválidos”.

9c.1 Si todos los campos no están llenos o hay caracteres inválidos, se vuelve a desplegar la página con el formulario y se marcan en rojo las letras del formulario de los campos inválidos. También se muestra un aviso indicando “Algunos campos son inválidos”.

**Caso de uso:** Iniciar sesión

**Actores participantes:** Administrador, cliente registrado

**Condiciones de entrada:** El usuario se ha registrado previamente y dispone de una contraseña, correo electrónico y permisos en la página.

**Flujo de eventos:**

1. Dentro de la página, el usuario selecciona la opción iniciar sesión.
2. El sistema solicita al usuario el correo electrónico y la contraseña.
3. El usuario introduce sus datos en los respectivos campos que le provee la página y selecciona la opción ingresar.
4. El sistema comprueba que dichos datos se encuentren en la base de datos.
5. El sistema inicia sesión con el usuario en cuestión.

**Condiciones de salida:** El usuario queda validado en el sistema, y se despliega la interfaz de la tienda.

**Flujos alternativos:**

5.1 Si los datos ingresados por el usuario no coinciden con los que se encuentran en la base de datos, el sistema muestra un mensaje advirtiendo que los datos son incorrectos o el usuario no se encuentra registrado.

**Caso de uso:** Buscar producto

**Actores participantes:** Administrador, cliente registrado, usuario no autenticado.

**Condiciones de entrada:** No tiene.

**Flujo de eventos:**

1. El usuario introduce el nombre (o parte de este) o código del producto que desea buscar en la barra de búsqueda que provee la página.
2. El usuario presiona enter o selecciona la opción buscar.
3. El sistema busca los productos que cumplen con el criterio indicado, y para cada producto encontrado provee la opción ver producto.

**Condiciones de salida:** El sistema muestra una lista con los productos encontrados en la base de datos que contienen el nombre o caracteres indicados.

**Flujos alternativos:**

* 1. Si el sistema no encuentra productos que cumplan con el criterio de búsqueda ingresado, entonces despliega un mensaje indicando que no se encontraron productos relacionados a los criterios de búsqueda.

**Caso de uso:** Ver producto

**Actores participantes:** Administrador, cliente registrado, usuario no autenticado.

**Condiciones de entrada:** El usuario debe haber ingresado en la opción buscar producto, el criterio de búsqueda que desea.

**Flujo de eventos:**

1. El usuario selecciona ver producto, opción que se despliega para cada producto que aparece en una búsqueda realizada.
2. El sistema obtiene los datos completos del producto seleccionado y los muestra en pantalla, así como también despliega la opción añadir a carrito.

**Condiciones de salida:** Se muestra en la página toda la información detallada del producto seleccionado.

**Flujos alternativos:** No tiene

**Caso de uso:** Añadir al carrito

**Actores participantes:** Cliente registrado, usuario no autenticado.

**Condiciones de entrada:** El usuario debe haber realizado la búsqueda de un producto y seleccionado la opción ver producto.

**Flujo de eventos:**

1. El usuario selecciona el botón "Añadir al carrito" que posee el producto en cuestión.
2. El sistema agrega el producto nuevo al carrito.
3. El sistema actualiza el contador de artículos y el total en el carrito (para que se vea actualizado por pantalla).

**Condiciones de salida:** El sistema agrega correctamente el producto al carrito virtual del cliente.

**Flujos alternativos:**

* 1. Si el producto ya existe en el carrito, se aumenta la cantidad que existe del mismo.

**Caso de uso:** Ver carrito

**Actores participantes:** Cliente registrado, usuario no autenticado.

**Condiciones de entrada:** No tiene.

**Flujo de eventos:**

1. El usuario selecciona la opción ver carrito.
2. El sistema comprueba la lista de productos agregados.
3. El sistema despliega una lista con todos los productos que han sido agregados al carrito por el usuario.
4. El sistema despliega la opción comprar/pagar y eliminar del carrito.

**Condiciones de salida:** Se muestra por pantalla todos los productos existentes en el carrito, con el total del carrito.

**Flujos alternativos:**

* 1. Si el usuario no ha agregado productos al carrito, el sistema muestra un mensaje indicando que el carrito de compras está vacío.

**Caso de uso:** Comprar/Pagar

**Actores participantes:** Cliente

**Condiciones de entrada:** El cliente debe estar previamente registrado en el sistema y haber iniciado sesión. También debe haber ingresado a la opción ver carrito y tener al menos un producto agregado en el carrito

**Flujo de eventos:**

1. El cliente selecciona la opción "Pagar".
2. El sistema obtiene el precio total de los productos que están en el carrito.
3. El sistema muestra por pantalla una factura virtual y los datos de la cuenta a la que debe depositar el cliente, además de un código único asociado a la compra realizada (Para llevar registro de la compra y en caso de reclamos).

**Condiciones de salida:** La compra queda registrada en la base de datos. El sistema muestra correctamente por pantalla los datos referentes a la compra.

**Flujos alternativos:**

* 1. Si el usuario no ha iniciado sesión como cliente al momento de seleccionar la opción pagar, se le pide que ingrese al sistema con sus datos o se registre, para así poder realizar la respectiva compra.

**Caso de uso:** Eliminar del carrito

**Actores participantes:** Cliente registrado, usuario no autenticado.

**Condiciones de entrada:** El usuario debe ingresar a la opción ver carrito. El producto debe haber sido añadido al carrito anteriormente.

**Flujo de eventos:**

1. El usuario selecciona la opción eliminar del carrito, correspondiente al producto que desea borrar.
2. El sistema elimina del carrito el producto seleccionado.
3. El sistema actualiza el total del carrito.

**Condiciones de salida:** Se muestra por pantalla el carrito actualizado, sin el producto que se eliminó.

**Flujos alternativos:** No tiene.

**Caso de uso:** Ver datos

**Actores participantes:** Cliente registrado, administrador.

**Condiciones de entrada:** El usuario debe estar registrado y haber iniciado sesión.

**Flujo de eventos:**

1. El usuario selecciona la opción ver datos, para poder ver sus respectivos datos que se encuentran registrados en la página.
2. El sistema muestra los datos actuales del usuario.

**Condiciones de salida:** Se despliega todos los datos e información almacenada en el sistema, del respectivo usuario.

**Flujos alternativos:** No tiene.

**Caso de uso:** Modificar datos.

**Actores participantes:** Administrador, cliente registrado.

**Condiciones de entrada:** El usuario debe haber iniciado sesión y seleccionar la opción ver datos.

**Flujo de eventos:**

1. El usuario selecciona la opción modificar datos.
2. El sistema despliega los campos con los datos almacenados, de manera editable, para que se puedan modificar.
3. El usuario modifica los datos que desea cambiar.
4. El usuario selecciona la opción guardar, para guardar los cambios realizados.
5. El sistema registra y actualiza los cambios realizados en la base de datos.

**Condiciones de salida:** Datos modificado en la base de datos del sistema.

**Flujos alternativos:**

* 1. El usuario cancela la operación de guardar los datos modificados.
  2. El sistema regresa a la interfaz donde se despliegan todos los datos del usuario, junto con la opción modificar datos.
  3. Si los datos introducidos para modificar, no son válidos, el sistema advierte el error y vuelve a mostrar el formulario.

**Caso de uso:** Gestionar Productos

**Actores participantes:** Administrador

**Condiciones de entrada:** El usuario debe iniciar sesión, realizando una identificación válida en el sistema como administrador.

**Flujo de eventos:**

1. El sistema despliega las opciones que le provee al administrador para administrar los productos, como: registrar producto y buscar producto.

**Condiciones de salida:** Se abren correctamente la sección de Productos.

**Flujos alternativos:** No tiene.

**Caso de uso:** Consultar productos

**Actores participantes:** Administrador

**Condiciones de entrada:** El usuario debe iniciar sesión, realizando una identificación válida en el sistema como administrador, he ingresar a la opción gestionar productos.

**Flujo de eventos:**

1. El administrador solicita al sistema la visualización de datos de producto.
2. El sistema muestra los resultados, permitiéndole al usuario ver la información detallada del producto que se está buscando.

**Condiciones de salida:** El sistema muestra todos los datos registrados del producto que se está buscando.

**Flujos alternativos:**

* 1. Si el sistema no encuentra el producto que se está buscando, entonces despliega un mensaje indicando que no se encontró el producto relacionado al criterio de búsqueda ingresado.

**Caso de uso:** Registrar producto

**Actores participantes:** Administrador

**Condiciones de entrada:** El usuario debe haber iniciado sesión como administrador, he ingresar a la opción gestionar productos.

**Flujo de eventos:**

1. Dentro de la opción gestionar productos, el administrador selecciona registrar producto.
2. La página despliega una serie de campos, en donde el administrador ingresa los datos solicitados por el sistema como:

- Código del producto

- Tipo

- Precio

- Modelo: Nombre, características.

1. Se elige la opción guardar para poder registrar el producto en el sistema.
2. El sistema guarda en la base de datos la información del producto ingresada por el administrador.

**Condiciones de salida:** El producto queda almacenado en el sistema.

**Flujos alternativos:**

* 1. El administrador cancela la operación de registrar el producto.
  2. El sistema regresa a la opción gestionar productos.
  3. Si los datos introducidos no son válidos o el producto ya existe, el sistema advierte sobre los posibles errores y vuelve a mostrar el formulario.

**Caso de uso:** Modificar productos.

**Actores participantes:** Administrador.

**Condiciones de entrada:** El usuario debe haber iniciado sesión como administrador, y realizar la búsqueda del producto que desea modificar.

**Flujo de eventos:**

1. El administrador selecciona la opción modificar datos.
2. El sistema despliega los campos con los datos del producto almacenados, de manera editable, para que se puedan modificar.
3. El administrador modifica los datos que desea cambiar.
4. El usuario selecciona la opción guardar, para guardar los cambios realizados.
5. El sistema registra y actualiza los cambios realizados en la base de datos.

**Condiciones de salida:** Datos modificados del producto en la base de datos del sistema.

**Flujos alternativos:**

* 1. El administrador cancela la operación de guardar los datos modificados del producto.
  2. El sistema regresa a la interfaz donde se despliegan todos los datos del producto, junto con la opción modificar productos.
  3. Si los datos introducidos para modificar, no son válidos, el sistema advierte el error y vuelve a mostrar el formulario.

**Caso de uso:** Eliminar producto

**Actores participantes:** Administrador

**Condiciones de entrada:** El usuario debe haber iniciado sesión como administrador, y realizar la búsqueda del producto que desea eliminar.

**Flujo de eventos:**

1. El administrador selecciona la opción eliminar producto, que se despliega en la búsqueda y vista del producto que se desea eliminar.
2. El sistema elimina el producto de la base de datos.

**Condiciones de salida:** El producto ya no se encuentra en el sistema.

**Flujos alternativos:** No tiene.

**Caso de uso:** Buscar/Ver compras

**Actores participantes:** Administrador

**Condiciones de entrada:** El usuario debe iniciar sesión, realizando una identificación válida en el sistema como administrador.

**Flujo de eventos:**

1. Dentro de la página, el administrador elige la opción buscar/ver compras.
2. El sistema muestra un campo de búsqueda de compras.
3. El administrador introduce los valores para los campos, como el Id de la compra y pulsa Buscar
4. El sistema muestra una serie de campos con los datos de la compra realizada por el cliente.

**Condiciones de salida:** El sistema despliega toda la información referente a la compra buscada.

**Flujos alternativos:**

* 1. Si el sistema no encuentra compras que cumplan con el criterio de búsqueda ingresado, entonces despliega un mensaje indicando que no se encontraron compras para dichos criterios de búsqueda.

**Caso de uso:** Gestionar cliente

**Actores participantes:** Administrador

**Condiciones de entrada:** El administrador debe iniciar sesión y seleccionar la opción “Buscar cliente” del panel de administrador.

**Flujo de eventos:**

1. El sistema muestra un campo de búsqueda de clientes.
2. El administrador introduce los valores para los campos, como la cédula del cliente y selecciona Buscar.
3. El sistema llama a la función buscarCliente(ci) y retorna la información del cliente.
4. El sistema muestra una serie de campos con los datos del cliente.
5. El administrador da clic en el botón “Editar cliente”.
6. El sistema llama a la función editarCliente(ci).
7. El sistema habilita la edición de los campos con los datos del cliente.
8. El administrador modifica los datos que necesita.
9. El administrador da clic en “Guardar”.
10. El sistema registra los nuevos datos.
11. El sistema despliega el mensaje “Edición exitosa”.
12. El sistema redirige al administrador a la pantalla con los datos del cliente actualizados.

**Condiciones de salida:** El sistema despliega toda la información actualizada del cliente.

**Flujos alternativos:**

* 1. Si el sistema no encuentra un cliente que cumpla con el criterio de búsqueda ingresado, entonces despliega un mensaje indicando que no se encontró un cliente con dicho criterio de búsqueda.

8.1 El administrador ingresó caracteres inválidos.

**Caso de uso:** Buscar/Ver cliente

**Actores participantes:** Administrador

**Condiciones de entrada:** El usuario debe iniciar sesión, realizando una identificación válida en el sistema como administrador.

**Flujo de eventos:**

1. Dentro de la página, el administrador elige la opción buscar/ver cliente.
2. El sistema muestra un campo de búsqueda de clientes.
3. El administrador introduce los valores para los campos, como la cédula del cliente y selecciona Buscar.
4. El sistema muestra una serie de campos con los datos del cliente.

**Condiciones de salida:** El sistema despliega toda la información y datos referentes al cliente buscado.

**Flujos alternativos:**

* 1. Si el sistema no encuentra un cliente que cumpla con el criterio de búsqueda ingresado, entonces despliega un mensaje indicando que no se encontró un cliente con dicho criterio de búsqueda.

**Caso de uso:** Eliminar cliente

**Actores participantes:** Administrador

**Condiciones de entrada:** El administrador debe iniciar sesión y seleccionar la opción “Buscar cliente” del panel de administrador.

**Flujo de eventos:**

1. El sistema muestra un campo de búsqueda de clientes.
2. El administrador introduce los valores para los campos, como la cédula del cliente y selecciona Buscar.
3. El sistema muestra una serie de campos con los datos del cliente.
4. El administrador da clic en el botón “Eliminar cliente”.
5. El sistema llama a la función eliminarCliente(ci).
6. Se despliega un mensaje preguntando “¿Eliminar cliente?”.
7. El administrador da clic en “Sí”.
8. El sistema borra de la base de datos los datos asociados al cliente.
9. El sistema despliega el mensaje “Cliente eliminado”.
10. El sistema redirige al administrador al campo de búsqueda de clientes.

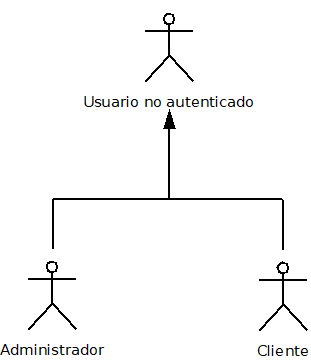
**Condiciones de salida:** El sistema redirige al administrador al campo de búsqueda de clientes después de la eliminación.

**Flujos alternativos:**

* 1. Si el sistema no encuentra un cliente que cumpla con el criterio de búsqueda ingresado, entonces despliega un mensaje indicando que no se encontró un cliente con dicho criterio de búsqueda.

7.1 El administrador da clic en “No” y se redirige al flujo 3.

**Jerarquía de actores:**



**Metodología a utilizar, roles de cada integrante del equipo:**

Para la realización del proyecto MakariónWeb, utilizaremos la metodología Scrum, asignando los roles a cada integrante del equipo de la siguiente manera:

1. Equipo de desarrollo:

* Génesis Del Mar Uzcategui Reinoza
* José Manuel Reinoza Duque

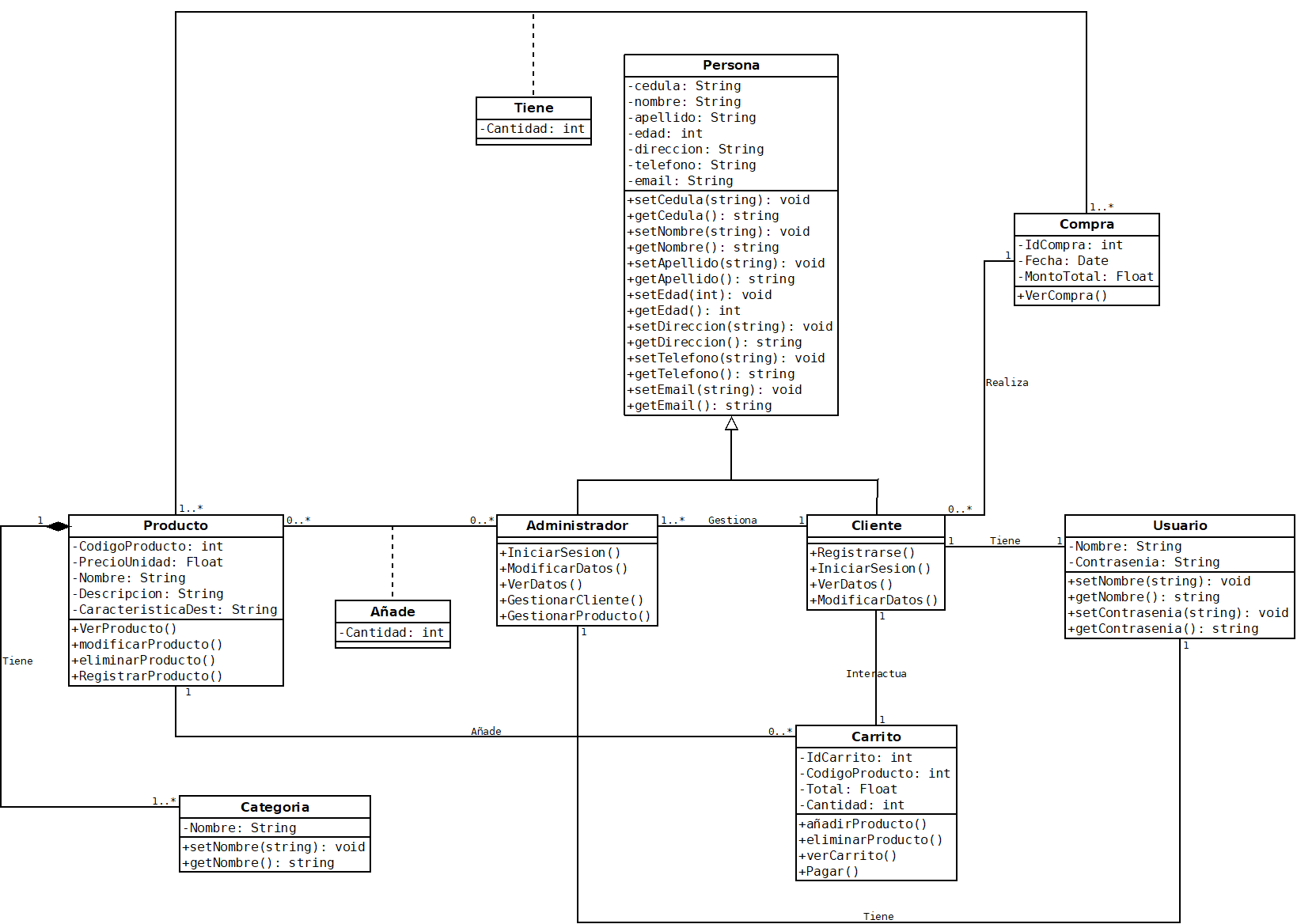
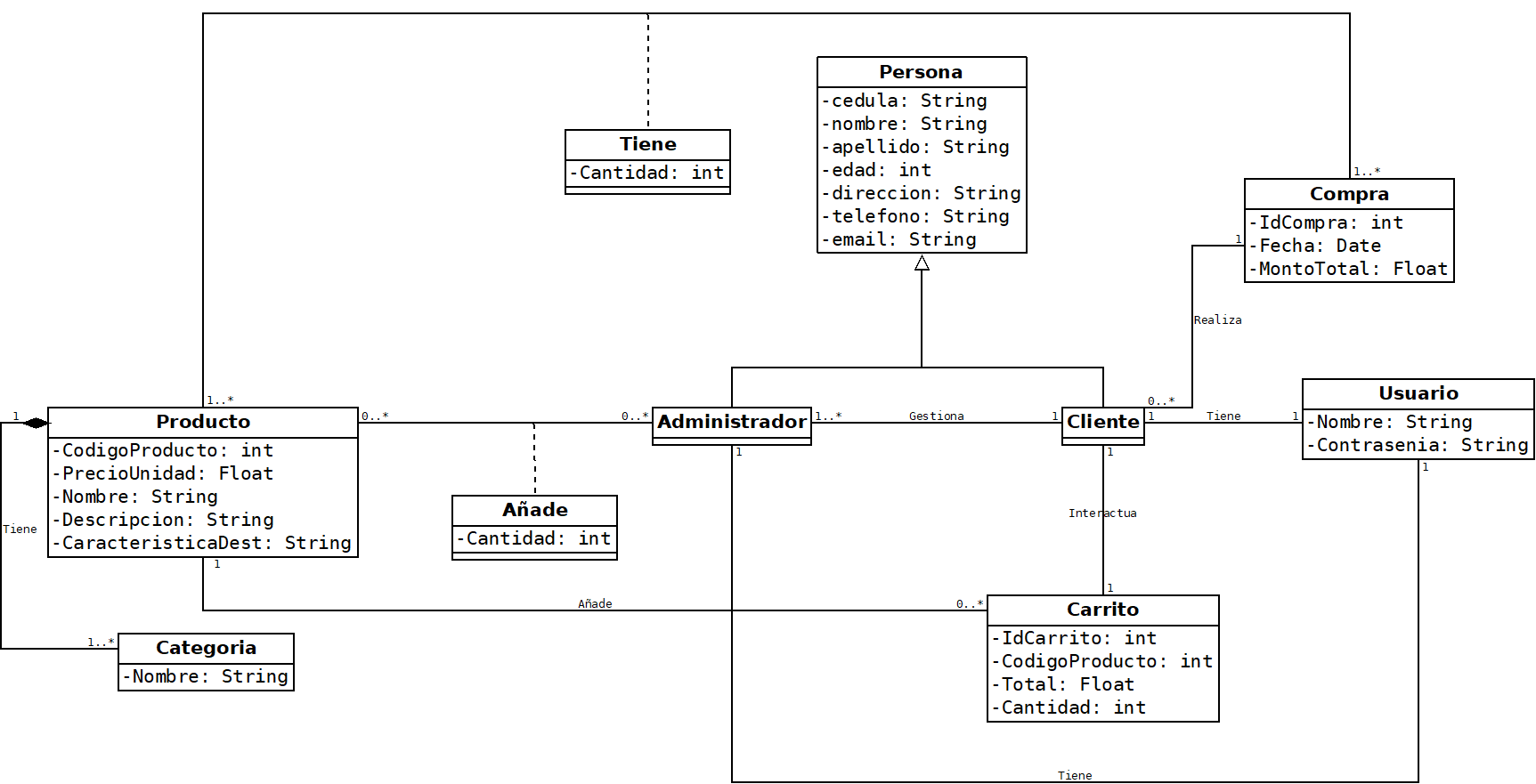
1. Product owner:

* Génesis Del Mar Uzcategui Reinoza

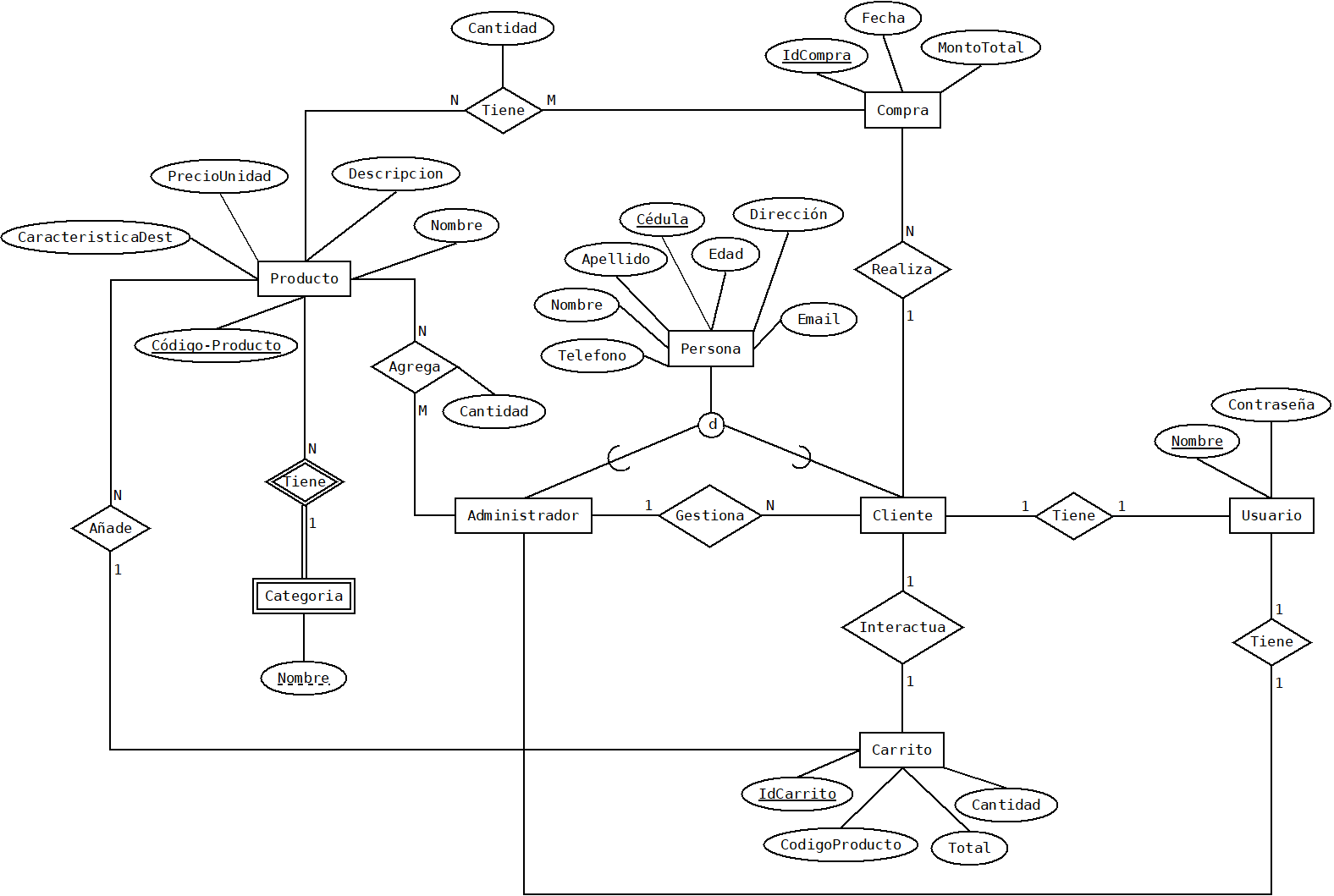
1. Scrum Master:

José Manuel Reinoza Duque

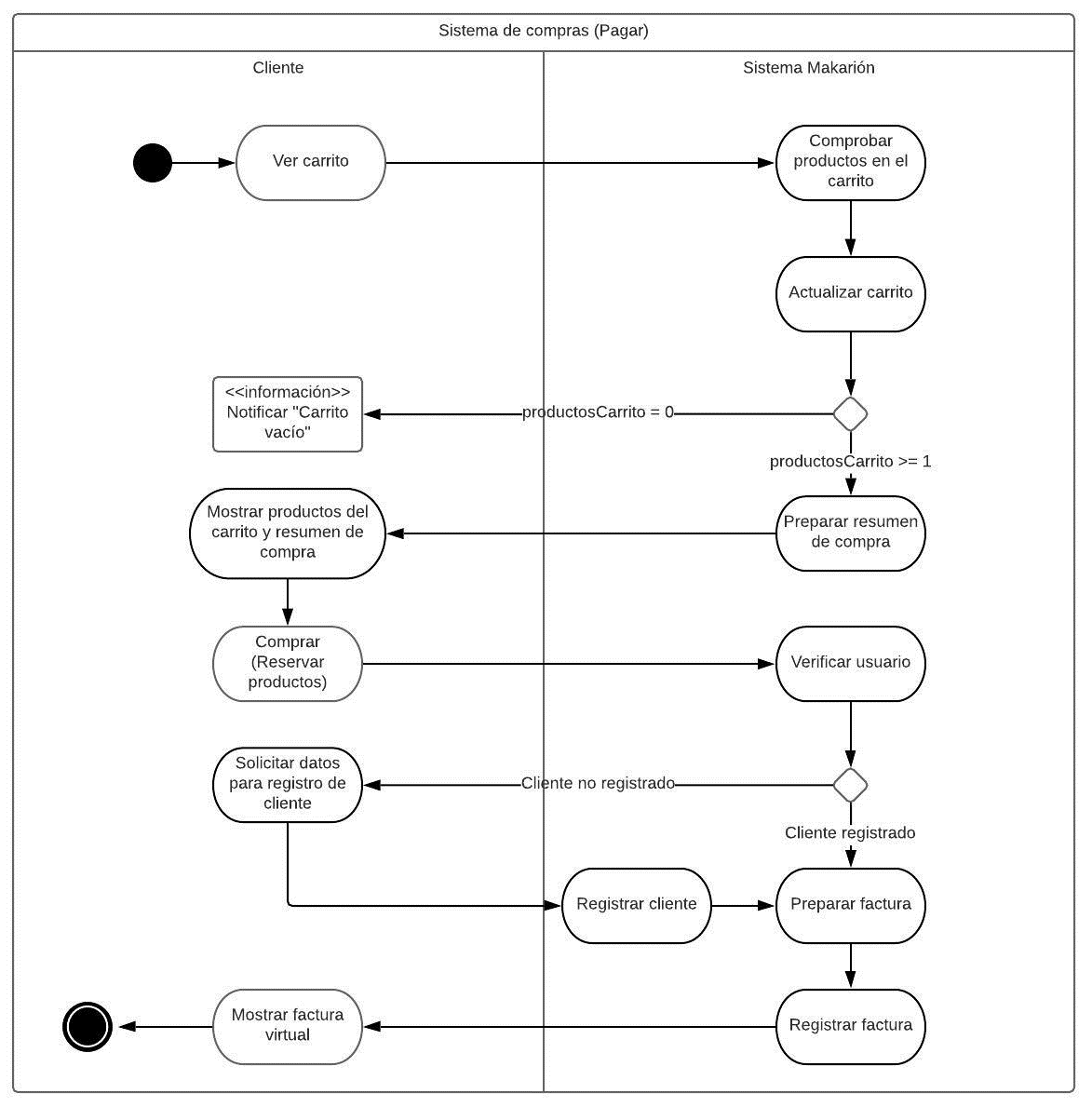
**Diagramas de clases**

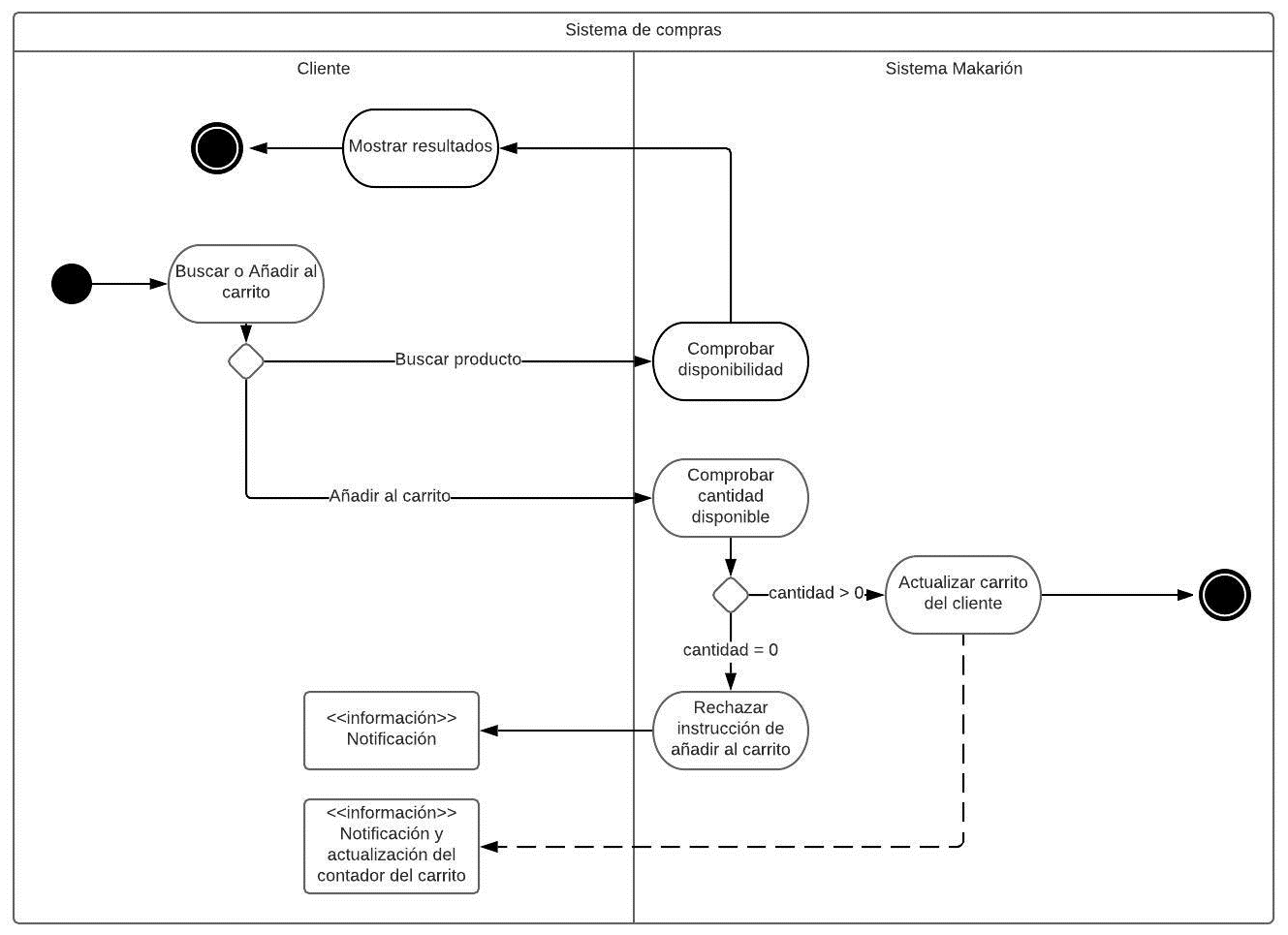


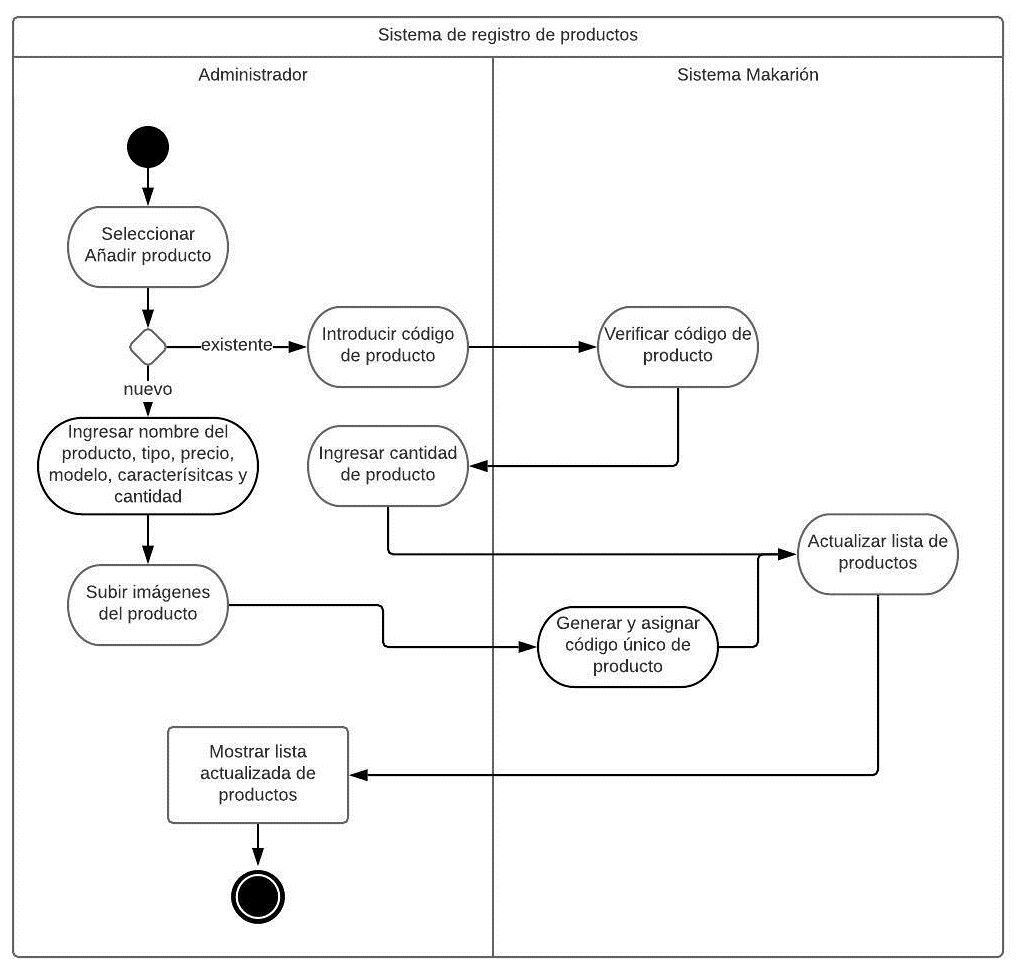
**Diagrama ERE**



**Diagramas de actividades**

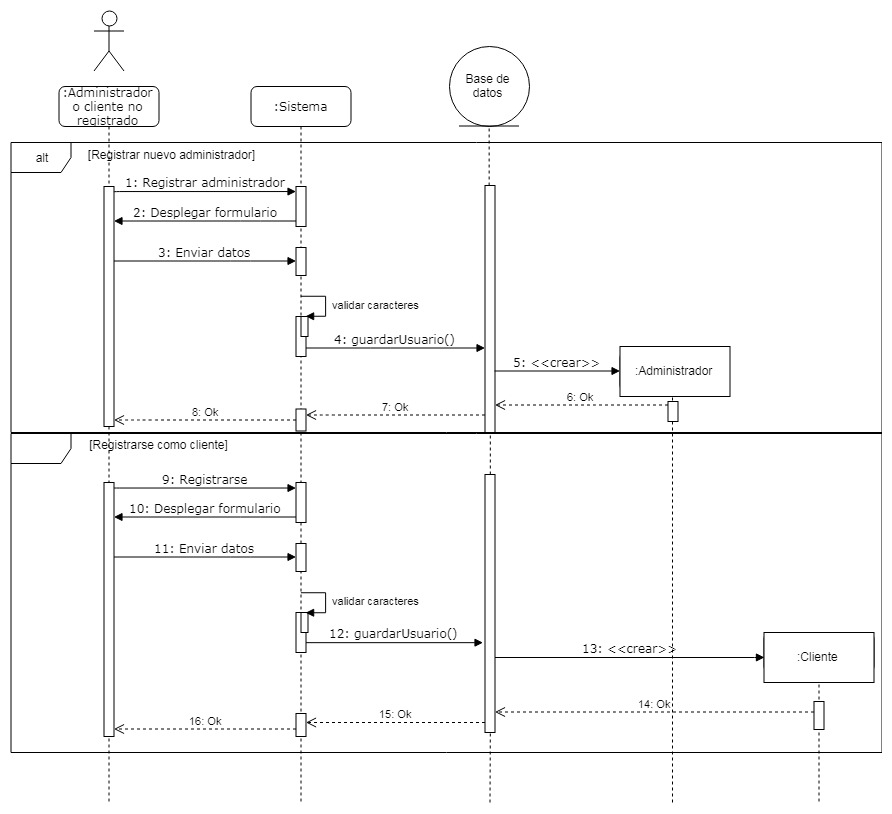
****

****

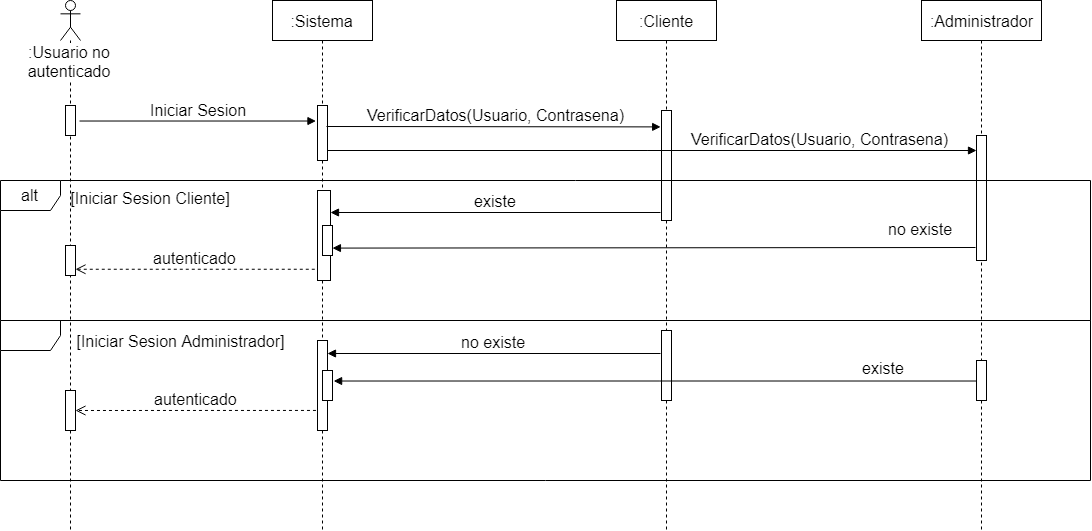


**Diagramas de secuencia**

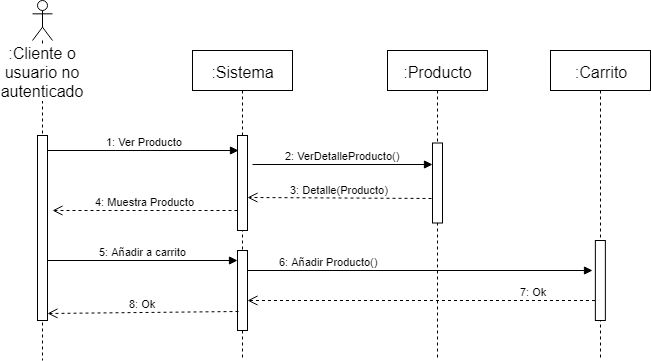
**- Registrarse**

****

**- Iniciar Sesión**



**- Ver producto, añadir al carrito**

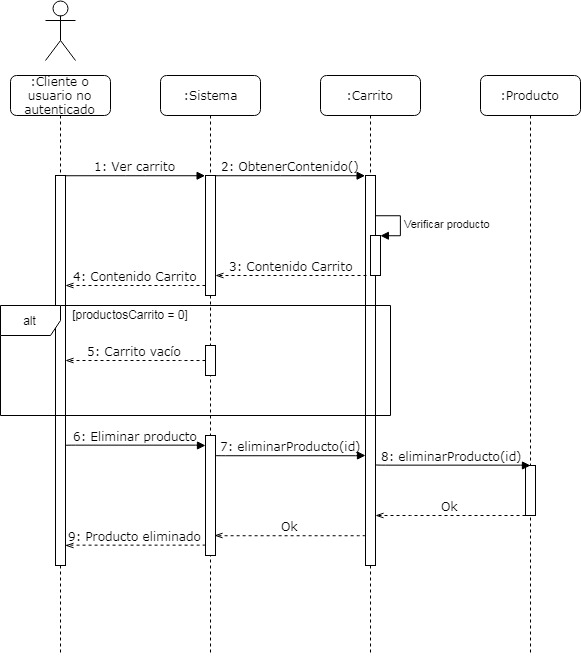


**- Comprar/Pagar**



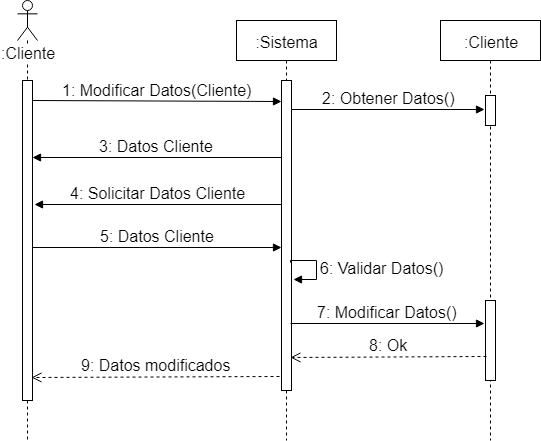
**- Eliminar del carrito**

:

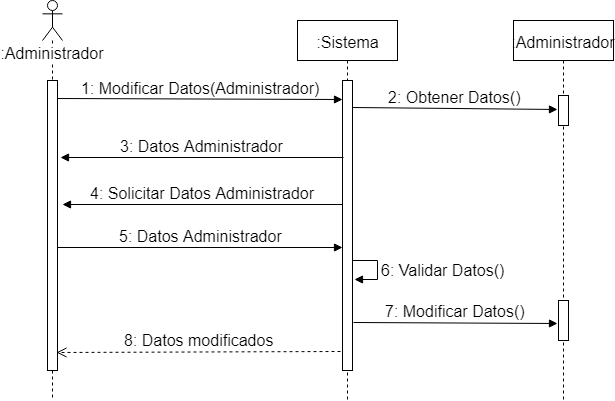
****

**- Modificar Datos**

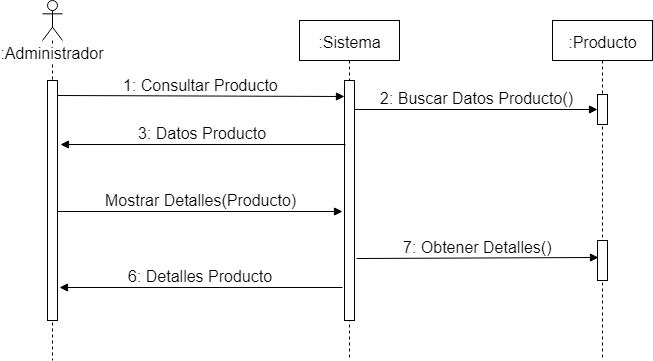
* **Como Cliente**



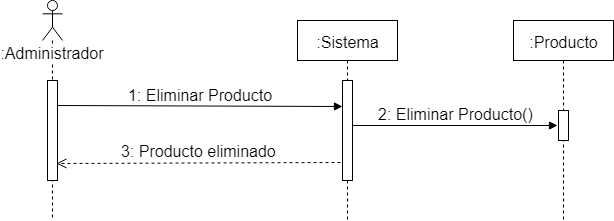
* **Como Administrador**



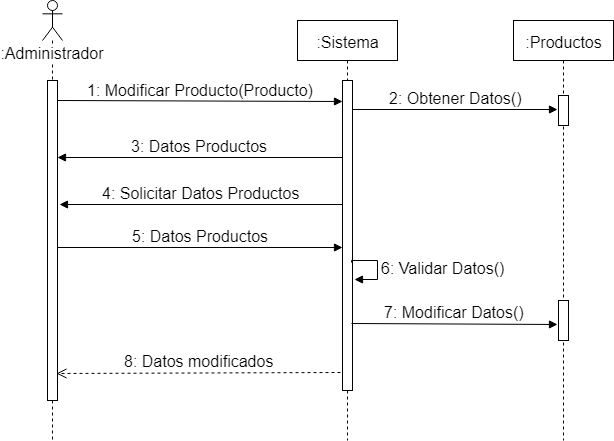
**- Consultar Producto**



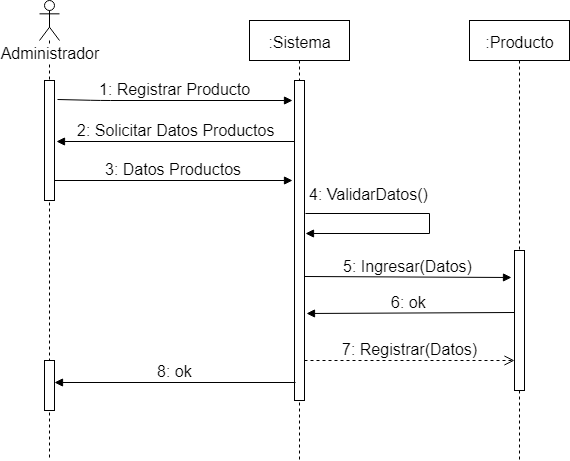
**- Eliminar Producto**



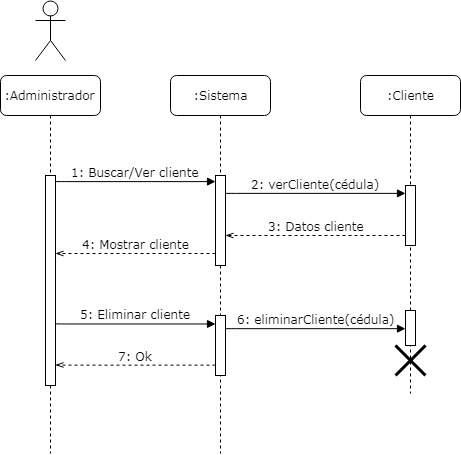
**- Modificar Producto**



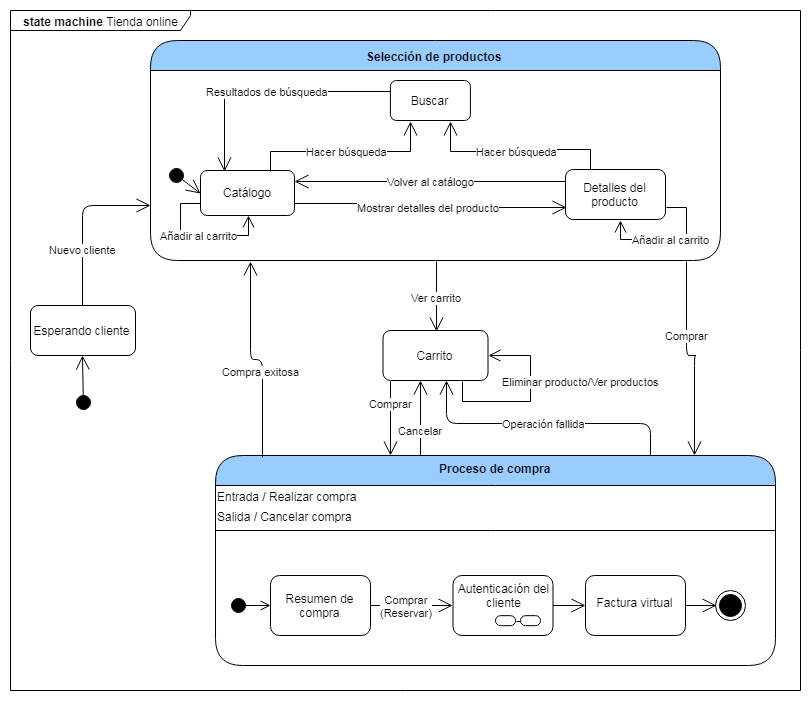
**- Registrar Producto**



**- Eliminar cliente**



**Diagrama de estado**



**Modelo en SQL:**

**- Carrito**

CREATE TABLE public.carrito\_carrito

(

id integer NOT NULL DEFAULT nextval('carrito\_carrito\_id\_seq'::regclass),

total numeric(18,2) NOT NULL,

"cantidadAgregada" integer NOT NULL,

producto\_id character varying(10) COLLATE pg\_catalog."default",

CONSTRAINT carrito\_carrito\_pkey PRIMARY KEY (id),

CONSTRAINT carrito\_carrito\_producto\_id\_e7c28d57\_fk\_producto\_ FOREIGN KEY (producto\_id)

REFERENCES public.producto\_producto (codigo) MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION

DEFERRABLE INITIALLY DEFERRED

)

WITH (

OIDS = FALSE

)

TABLESPACE pg\_default;

ALTER TABLE public.carrito\_carrito

OWNER to postgres;

-- Index: carrito\_carrito\_producto\_id\_e7c28d57

-- DROP INDEX public.carrito\_carrito\_producto\_id\_e7c28d57;

CREATE INDEX carrito\_carrito\_producto\_id\_e7c28d57

ON public.carrito\_carrito USING btree

(producto\_id COLLATE pg\_catalog."default")

TABLESPACE pg\_default;

-- Index: carrito\_carrito\_producto\_id\_e7c28d57\_like

-- DROP INDEX public.carrito\_carrito\_producto\_id\_e7c28d57\_like;

CREATE INDEX carrito\_carrito\_producto\_id\_e7c28d57\_like

ON public.carrito\_carrito USING btree

(producto\_id COLLATE pg\_catalog."default" varchar\_pattern\_ops)

TABLESPACE pg\_default;

**- Compra**

CREATE TABLE public.compra\_compra

(

"idCompra" character varying(10) COLLATE pg\_catalog."default" NOT NULL,

fecha date NOT NULL,

"MontoTotal" numeric(18,2) NOT NULL,

CONSTRAINT compra\_compra\_pkey PRIMARY KEY ("idCompra")

)

WITH (

OIDS = FALSE

)

TABLESPACE pg\_default;

ALTER TABLE public.compra\_compra

OWNER to postgres;

-- Index: compra\_compra\_idCompra\_87e5eafc\_like

-- DROP INDEX public."compra\_compra\_idCompra\_87e5eafc\_like";

CREATE INDEX "compra\_compra\_idCompra\_87e5eafc\_like"

ON public.compra\_compra USING btree

("idCompra" COLLATE pg\_catalog."default" varchar\_pattern\_ops)

TABLESPACE pg\_default;

**- Compra\_Producto**

CREATE TABLE public.compra\_compra\_producto

(

id integer NOT NULL DEFAULT nextval('compra\_compra\_producto\_id\_seq'::regclass),

compra\_id character varying(10) COLLATE pg\_catalog."default" NOT NULL,

producto\_id character varying(10) COLLATE pg\_catalog."default" NOT NULL,

CONSTRAINT compra\_compra\_producto\_pkey PRIMARY KEY (id),

CONSTRAINT compra\_compra\_producto\_compra\_id\_producto\_id\_31d6c71f\_uniq UNIQUE (compra\_id, producto\_id),

CONSTRAINT compra\_compra\_produc\_compra\_id\_7fbac5b6\_fk\_compra\_co FOREIGN KEY (compra\_id)

REFERENCES public.compra\_compra ("idCompra") MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION

DEFERRABLE INITIALLY DEFERRED,

CONSTRAINT compra\_compra\_produc\_producto\_id\_36e1f151\_fk\_producto\_ FOREIGN KEY (producto\_id)

REFERENCES public.producto\_producto (codigo) MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION

DEFERRABLE INITIALLY DEFERRED

)

WITH (

OIDS = FALSE

)

TABLESPACE pg\_default;

ALTER TABLE public.compra\_compra\_producto

OWNER to postgres;

-- Index: compra\_compra\_producto\_compra\_id\_7fbac5b6

-- DROP INDEX public.compra\_compra\_producto\_compra\_id\_7fbac5b6;

CREATE INDEX compra\_compra\_producto\_compra\_id\_7fbac5b6

ON public.compra\_compra\_producto USING btree

(compra\_id COLLATE pg\_catalog."default")

TABLESPACE pg\_default;

-- Index: compra\_compra\_producto\_compra\_id\_7fbac5b6\_like

-- DROP INDEX public.compra\_compra\_producto\_compra\_id\_7fbac5b6\_like;

CREATE INDEX compra\_compra\_producto\_compra\_id\_7fbac5b6\_like

ON public.compra\_compra\_producto USING btree

(compra\_id COLLATE pg\_catalog."default" varchar\_pattern\_ops)

TABLESPACE pg\_default;

-- Index: compra\_compra\_producto\_producto\_id\_36e1f151

-- DROP INDEX public.compra\_compra\_producto\_producto\_id\_36e1f151;

CREATE INDEX compra\_compra\_producto\_producto\_id\_36e1f151

ON public.compra\_compra\_producto USING btree

(producto\_id COLLATE pg\_catalog."default")

TABLESPACE pg\_default;

-- Index: compra\_compra\_producto\_producto\_id\_36e1f151\_like

-- DROP INDEX public.compra\_compra\_producto\_producto\_id\_36e1f151\_like;

CREATE INDEX compra\_compra\_producto\_producto\_id\_36e1f151\_like

ON public.compra\_compra\_producto USING btree

(producto\_id COLLATE pg\_catalog."default" varchar\_pattern\_ops)

TABLESPACE pg\_default;

**- Producto**

CREATE TABLE public.producto\_producto

(

codigo character varying(10) COLLATE pg\_catalog."default" NOT NULL,

nombre character varying(50) COLLATE pg\_catalog."default" NOT NULL,

categoria character varying(10) COLLATE pg\_catalog."default" NOT NULL,

precio numeric(18,2) NOT NULL,

precio\_en\_descuento numeric(18,2),

descripcion text COLLATE pg\_catalog."default" NOT NULL,

cantidad integer NOT NULL,

disponible boolean NOT NULL,

unidades\_vendidas integer NOT NULL,

creado\_el timestamp with time zone NOT NULL,

actualizado\_el timestamp with time zone NOT NULL,

imagen character varying(100) COLLATE pg\_catalog."default",

imagen\_2 character varying(100) COLLATE pg\_catalog."default",

imagen\_3 character varying(100) COLLATE pg\_catalog."default",

imagen\_4 character varying(100) COLLATE pg\_catalog."default",

imagen\_5 character varying(100) COLLATE pg\_catalog."default",

texto\_destacado character varying(100) COLLATE pg\_catalog."default" NOT NULL,

caracteristica\_destacada character varying(60) COLLATE pg\_catalog."default",

caracteristica\_destacada\_2 character varying(60) COLLATE pg\_catalog."default",

caracteristica\_destacada\_3 character varying(60) COLLATE pg\_catalog."default",

caracteristica\_destacada\_4 character varying(60) COLLATE pg\_catalog."default",

caracteristica\_destacada\_5 character varying(60) COLLATE pg\_catalog."default",

CONSTRAINT producto\_producto\_pkey PRIMARY KEY (codigo),

CONSTRAINT producto\_producto\_unidades\_vendidas\_check CHECK (unidades\_vendidas >= 0)

)

WITH (

OIDS = FALSE

)

TABLESPACE pg\_default;

ALTER TABLE public.producto\_producto

OWNER to postgres;

-- Index: producto\_producto\_codigo\_8e46d840\_like

-- DROP INDEX public.producto\_producto\_codigo\_8e46d840\_like;

CREATE INDEX producto\_producto\_codigo\_8e46d840\_like

ON public.producto\_producto USING btree

(codigo COLLATE pg\_catalog."default" varchar\_pattern\_ops)

TABLESPACE pg\_default;

**- Administrador y Cliente:** Como ambos acceden al sistema y tienen los mismos atributos, entonces se utilizó el user de django, por lo que se genera:

* **users\_customuser**

CREATE TABLE public.users\_customuser

(

id integer NOT NULL DEFAULT nextval('users\_customuser\_id\_seq'::regclass),

password character varying(128) COLLATE pg\_catalog."default" NOT NULL,

last\_login timestamp with time zone,

is\_superuser boolean NOT NULL,

username character varying(150) COLLATE pg\_catalog."default" NOT NULL,

first\_name character varying(30) COLLATE pg\_catalog."default" NOT NULL,

last\_name character varying(150) COLLATE pg\_catalog."default" NOT NULL,

email character varying(254) COLLATE pg\_catalog."default" NOT NULL,

is\_staff boolean NOT NULL,

is\_active boolean NOT NULL,

date\_joined timestamp with time zone NOT NULL,

CONSTRAINT users\_customuser\_pkey PRIMARY KEY (id),

CONSTRAINT users\_customuser\_username\_key UNIQUE (username)

)

WITH (

OIDS = FALSE

)

TABLESPACE pg\_default;

ALTER TABLE public.users\_customuser

OWNER to postgres;

-- Index: users\_customuser\_username\_80452fdf\_like

-- DROP INDEX public.users\_customuser\_username\_80452fdf\_like;

CREATE INDEX users\_customuser\_username\_80452fdf\_like

ON public.users\_customuser USING btree

(username COLLATE pg\_catalog."default" varchar\_pattern\_ops)

TABLESPACE pg\_default;

* **users\_completeuser**

CREATE TABLE public.users\_completeuser

(

customuser\_ptr\_id integer NOT NULL,

cedula character varying(10) COLLATE pg\_catalog."default" NOT NULL,

edad integer NOT NULL,

direccion text COLLATE pg\_catalog."default" NOT NULL,

telefono character varying(12) COLLATE pg\_catalog."default" NOT NULL,

CONSTRAINT users\_completeuser\_pkey PRIMARY KEY (cedula),

CONSTRAINT users\_completeuser\_customuser\_ptr\_id\_key UNIQUE (customuser\_ptr\_id)

,

CONSTRAINT users\_completeuser\_customuser\_ptr\_id\_79b7918b\_fk\_users\_cus FOREIGN KEY (customuser\_ptr\_id)

REFERENCES public.users\_customuser (id) MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION

DEFERRABLE INITIALLY DEFERRED

)

WITH (

OIDS = FALSE

)

TABLESPACE pg\_default;

ALTER TABLE public.users\_completeuser

OWNER to postgres;

CREATE INDEX users\_completeuser\_cedula\_2577efa5\_like

ON public.users\_completeuser USING btree

(cedula COLLATE pg\_catalog."default" varchar\_pattern\_ops)

TABLESPACE pg\_default;

**Arquitectura de software seleccionada para el proyecto**

Para el proyecto, se utilizará la arquitectura Cliente-Servidor, que se encuentra basada en el paso de mensajes entre una maquina (cliente) que solicita peticiones de servicio a otra en la que residen los datos y los programas de aplicación (servidor).



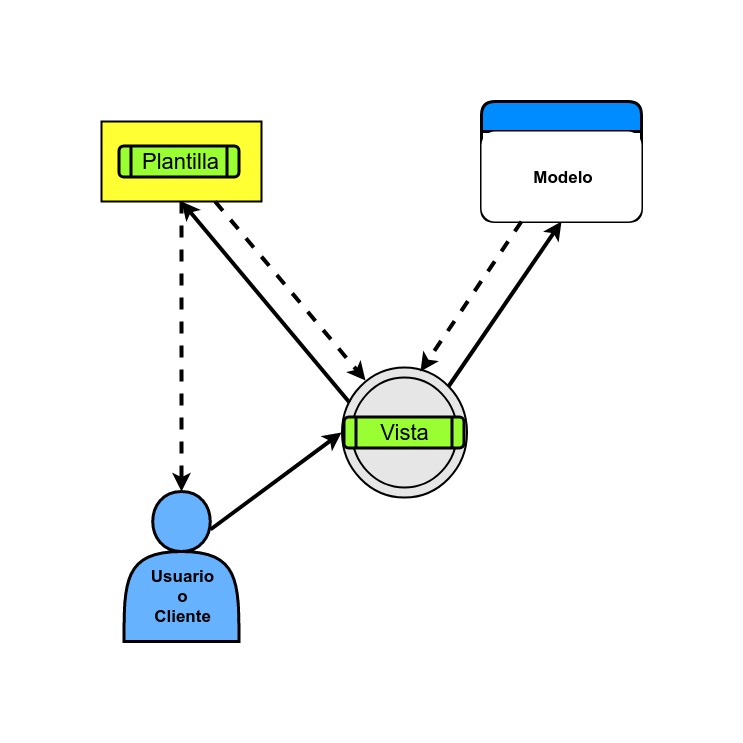
Conjuntamente se implementara también la arquitectura Modelo-Vista-Controlador (MVC), pero con algunas modificaciones, quedando como Modelo-Plantilla-Vista o Model - Template – View (MTV), esto debido al framework que se utilizara en la elaboración de este proyecto, el cual es Django.

La arquitectura MVC se basa en:

* **M**: “Model” (Modelo), la cual es una capa de abstracción en código de la base de datos manejada como un objeto a través del framework, el cual se encarga de la comunicación e interacción con la base de datos.
* **V**: “View” (Vista), donde se selecciona la manera en la que se presentan y muestran los datos extraídos de la base de datos, a través del modelo, que hayan sido requeridos por el controlador.
* **C**: “Controller” (Controlador), la sección sobre la cual se suele realizar la mayor parte del trabajo de Backend (la lógica de negocio), es el intermediario entre las solicitudes de usuario (cliente) a través de la vista y los modelos. Se encarga de designar qué datos deben ser pedidos al modelo y mostrados en la vista.

Debido a que en el MVC, la **C** en Django es manejada por el mismo framework y la parte más importante se produce en los modelos, las plantillas y las vistas, entonces utilizaremos MTV donde:

* **M**: “Model” (Modelo), sigue siendo la capa designada a la interacción y comunicación, con la información alojados en la base de datos.
* **T:** "Template" (Plantilla), la capa de presentación. Esta capa contiene las decisiones relacionadas a la presentación: como algunas cosas son mostradas sobre una página web u otro tipo de documento.
* **V:** “View” (Vista), la “C” en el patrón de diseño MTV, es la capa designada a la lógica de negocios, a través de la que pasarán los datos del modelo a la plantilla.



**Tecnología a utilizar y descripción de plataforma de desarrollo**

**Django**

Es un [framework](https://es.wikipedia.org/wiki/Framework" \o "Framework) de desarrollo web de [código abierto](https://es.wikipedia.org/wiki/Open_Source), escrito en [Python](https://es.wikipedia.org/wiki/Python), que respeta el [Modelo–vista–template](https://docs.djangoproject.com/en/2.0/), explicado anteriormente en la arquitectura seleccionada.

Para la creación de un sitio web, se necesita un conjunto de componentes similares: una manera de manejar la autenticación de usuarios (registrarse, iniciar sesión, cerrar sesión), un panel de administración para el sitio web, formularios, una forma de subir archivos, etc.

La meta fundamental de Django es facilitar la creación de todos estos componentes, permitiendo desarrollar sitios web complejos. Django pone énfasis en el re-uso, la conectividad y extensibilidad de componentes, el desarrollo rápido y el principio “[No te repitas](https://es.wikipedia.org/wiki/No_te_repitas)” (DRY, del inglés Don't Repeat Yourself). Python es usado en todas las partes del framework, incluso en configuraciones, archivos, y en los modelos de datos.

**HTML5**

El lenguaje HTML (*HyperText Markup Language*) se usa para la creación de páginas web en Internet, HTML es un lenguaje que permite definir documentos de hipertexto a base de ciertas etiquetas que marcan partes del documento dándoles una estructura o jerarquía, y que permite presentar el texto de una manera ordenada y agradable, con enlaces que conducen a otros documentos o fuentes de información relacionadas, y con inserciones multimedia.

**CSS3**

Cascading Style Sheets que se suele traducir por hojas de estilo en cascada es un lenguaje usado para para definir y crear la presentación de un documento estructurado escrito en un [lenguaje de marcado](https://es.wikipedia.org/wiki/Lenguaje_de_marcado). ​ Es muy usado para establecer el diseño visual de los documentos web, e interfaces de usuario escritas en [HTML](https://es.wikipedia.org/wiki/HTML), entre otros.

CSS está diseñado principalmente para marcar la separación del contenido del documento y la forma de presentación de este, características tales como las capas o layouts, los colores y las fuentes. Esta separación busca mejorar la accesibilidad del documento, proveer más flexibilidad y control en la especificación de características presentacionales, permitir que varios documentos HTML compartan un mismo estilo usando una sola hoja de estilos separada en un archivo .css, y reducir la complejidad y la repetición de código en la estructura del documento.

**JavaScript**

Javascript es un lenguaje de programación interpretado. Se define como orientado a objetos, débilmente tipado y dinámico.

Se utiliza principalmente en su forma del lado del cliente, implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas.

JavaScript se diseñó con una sintaxis similar al lenguaje C, aunque adopta nombres y convenciones del lenguaje de programación Java.

Todos los navegadores modernos interpretan el código JavaScript integrado dentro de las páginas web. Para interactuar con una página web se provee al lenguaje JavaScript de una implementación del *Document Object Model* (DOM).

**jQuery**

Permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones y agregar interacción con la técnica AJAX a las páginas de la tienda.

**PostgreSQL**

Es un [sistema de gestión de bases de datos](https://es.wikipedia.org/wiki/Sistema_de_gesti%C3%B3n_de_bases_de_datos) [relacional](https://es.wikipedia.org/wiki/Modelo_relacional) [orientado a objetos](https://es.wikipedia.org/wiki/Base_de_datos_orientada_a_objetos) y de [código abierto](https://es.wikipedia.org/wiki/Software_de_c%C3%B3digo_abierto), publicado bajo la [licencia PostgreSQL](https://es.wikipedia.org/w/index.php?title=Licencia_PostgreSQL&action=edit&redlink=1).

Una base de datos relacional almacena los datos en tablas separadas, lo cual hace que la velocidad y flexibilidad de trabajo sean muy grandes. Cada tabla esta enlazada mediante relaciones, con lo cual es posible combinar datos de varias tablas cuando se realiza una consulta.

Una característica interesante de PostgreSQL es el control de concurrencias multiversión o MVCC por sus siglas en inglés. Este método agrega una imagen del estado de la base de datos a cada transacción. Esto permite hacer transacciones eventualmente consistentes, ofreciendo grandes ventajas en el rendimiento.

Para el manejo de PostgreSQL, utilizaremos [**pgAdmin**](http://www.pgadmin.org/), que es la herramienta oficial para administrar nuestras bases de datos en PostgreSQL. Nos permite desde hacer búsquedas SQL hasta desarrollar toda nuestra base de datos de manera más fácil e intuitiva; directamente desde la interfaz gráfica.

**Bootstrap**

Es de gran utilidad para el desarrollo [front-end](https://en.wikipedia.org/wiki/Front-end_web_development" \o "en:Front-end web development), pues contiene plantillas de diseño con tipografía, formularios, botones, cuadros, menús de navegación y otros elementos de diseño basado en [HTML](https://es.wikipedia.org/wiki/HTML) y [CSS](https://es.wikipedia.org/wiki/Hojas_de_estilo_en_cascada), así como extensiones de [JavaScript](https://es.wikipedia.org/wiki/JavaScript) adicionales.

**Estilos de programación**

***Estilo de variables y funciones***

Para el nombrado de las variables y funciones se utilizará el estilo camelCase, por ejemplo: facturarCompra(), anadirAlCarrito(), idDelProducto, cantidadDisponible.

Todos los nombres siempre empezarán con una letra.

Nunca se usarán guiones debido a que no están permitidos en los nombres de JavaScript, puesto que puede reconocerlos como una operación de sustracción. Ejemplo no permitido para declarar una variable: factura-virtual.

***Estilo de identación***

Siempre se usarán 2 espacios para la identación de bloques en lugar de tabulación.

function facturarCompra(...) {  
  return (factura);  
}

***Estilo para las llaves***

Como convención para las llaves de las estructuras de control, se dejará la llave de apertura en la misma línea donde se declara la estructura. En caso de que exista una decisión después de una llave de cierre, se empezará la decisión en la misma línea después de la llave de cierre, ejemplo:

if (…) {

//…

} else {

do {

//…

} while (…)

//…

}

***Estilo del código en general***

Aunque en HTML y JavaScript no es indispensable usar **;** para finalizar una instrucción, en nuestro código siempre finalizaremos las instrucciones con **;**

De igual modo, siempre usaremos la palabra reservada **var** para declarar variables, aunque no sea obligatorio en JavaScript.

Se usarán espacios entre operadores y después de comas para mejorar la legibilidad del código. Ejemplo:

var precioTotal = precio \* cantidad;  
var nombreProductos = ["Bolso de mano", "Cartera de tela"];

***Estilo para la definición de objetos***

* La llave de apertura se dejará en la misma línea del nombre del objeto.
* Usaremos dos puntos seguido de un espacio para cada atributo y su valor.
* No se usará coma después del último atributo-valor del objeto.
* La llave de cierre se dejará en una nueva línea y sin espacios.
* Se terminarán las definiciones de objetos con un punto y coma después de la llave de cierre.

var usuario = {  
  nombre: "Maribel",  
  apellido: "Torres",

cedula: 18765331  
  nombreDeUsuario: "MariT18",  
  contrasena: m4ribelT0rs,  
 tipoDeUsuario: "Cliente"

};

***Nombres de archivos***

Los nombres de archivos externos, por ejemplo bolso.jpg, se escribirán completamente en minúsculas.

**Enlace al repositorio en GitHub**

<https://github.com/REI94/InventoryMakarion>

**Mockups:**

