

Taller de Programación 2

### Analista de Sistemas Curso 2°1° “C” Tomás Florimo Matías Siminelakis

“Sistema de gestión de compra de insumos electrónicos”

### “S.I.G.E.”

Documentación:

Este programa es de venta de productos la cual tiene interacción con los cliente ,este programa interactúa con sus usuario a través de peticiones con la cual se genera movimiento de datos a través de un sistema que conecta la base de datos por ruta con el usuario ,estas rutas tienen métodos que pueden generar cambios o simplemente mostrar los datos,aunque hay condiciones que debe complir de lo contrario mostrara un mensaje de error,una de las condiciones pueden ser de autenticación ya que no todos los usuario tienen el mismo cargo para interactuar con los datos todo esto se basa en un servidor r lo que hace posible todas estas interacciones de los usuarios con la base de dato ,tambien cuentan con una lógica interna que tiene funcionalidades de generar accesos a ciertos tipo de usuario a través de métodos internos que verifican el tipo de rol que aparece en los datos del usuario ,generar descuentos segun su metodo de pago y la cantidad de producto a la hora de hacer la compra..

Objetivo

El objetivo de este sistema a desarrollar, es brindar la posibilidad de darle a una empresa y a sus clientes la posibilidad de interactuar a través de un e-commerce. De esta manera, la empresa tiene una mejor gestión de los productos exhibidos para la compra, y los clientes tienen la posibilidad de comprar de manera online en el local, desde la comodidad de sus casas.

Funcionalidades solicitadas

* Visión de lista de productos
* Registración de cuenta, login y logout
* ABM total de productos en stock
* Comprar un producto indicando un medio de pago

Reglas de negocio

- Se espera que los clientes puedan visualizar los productos sin haberse registrado en la página, pero de manera que no puedan comprar.

- Un cliente no puede comprar sin haber iniciado sesión en el sitio.

- Un cliente podrá listar todos los productos y obtenerlos según su categoría, estando registrado o no.

- Los administradores (con credenciales validadas) serán quienes carguen los productos y categorías, y quienes administren al resto de usuarios.

- Se procesa el pago y hace una validación de la cantidad de productos que se compraron, y del método de pago que se utilizó, en base a los resultados de cada validación, se aplica un descuento (o no).

Casos de uso

| CUXX - Nombre | CU01 - Alta, y modificación de clientes |
| --- | --- |
| Descripción | Ingreso de datos de cliente en el sistema |
| Actores | Clientes |
| Precondición | El cliente no debe estar registrado en el sitio. |
| Resumen | A través de un formulario de altas, el cliente se puede registrar en la página. La información del cliente que se incluirá en dicho formulario será: el nombre y apellido, el teléfono y la fecha de nacimiento. Del mismo modo puede acceder a su perfil y alterar sus propios datos una vez registrado. |
| Postcondición | Se ingresa/actualiza el cliente en el sistema. |
| Observaciones | La próxima vez que el cliente intente acceder a este panel podrá editar sus datos. |

| CUXX - Nombre | CU02 - Logueo de cliente |
| --- | --- |
| Descripción | Autenticación del estado de logueo del cliente |
| Actores | Clientes |
| Precondición | El cliente debe intentar interactuar con el sistema y debe estar registrado. |
| Resumen | El cliente va a interactuar con la lista de productos. Si está logueado en el sistema podrá comprar, caso contrario no podrá comprar. |
| Postcondición | En el primer caso, el sistema envía una alerta al cliente, y en el segundo caso el cliente puede entrar en el sistema |
| Observaciones | Ninguna. |

User Stories CU01 y CU02:

* Como cliente quiero poder ingresar a la página y ser recibido en una landing page.
* Como cliente quiero poder abrir el apartado de productos y ver los detalles de los productos.
* Como cliente quiero poder registrarme para poder elegir productos y hacer compras.
* Como cliente sin loguear, al hacer clic en el botón de compra debería ser redireccionado a una pantalla de login/registro.

| CUXX - Nombre | CU03 - Listado de productos |
| --- | --- |
| Descripción | El sistema lista todos los productos en stock |
| Actores | Usuario |
| Precondición | El usuario puede estar logueado o no |
| Resumen | Independientemente del estado de logueo del usuario, se van a listar todos los productos según categoría y orden que solicite el usuario. La única variante es la validación de la autenticación del usuario. Si está autorizado puede comprar, caso negativo será redireccionado al login. |
| Postcondición | En el primer caso, el sistema envía una alerta al cliente, y en el segundo caso el cliente puede comprar un producto de la lista con stock |
| Observaciones | Ninguna. |

| CUXX - Nombre | CU04 - Búsqueda de un Producto |
| --- | --- |
| Descripción | Mostrar un producto especificado |
| Actores | Usuario |
| Precondición | El Producto debe estar registrado |
| Resumen | El cliente va a interactuar con la lista de productos. Si el producto esta en el sistema podrá verlo, caso contrario no podrá comprar. |
| Postcondición | En el primer caso el sistema le muestra el producto,y en el segundo caso el sistema envía un error al cliente. |
| Observaciones | Ninguna. |

User Story 03:

-como cliente quiero que aparezca la categorias de los productos en stock

-Como usuario quiero poder saber el nombre de todos los productos con su cantidad de stock..

User Story 04:

-Como cliente quiero ver un producto especifico.

-Como usuario quiero que aparezca solo si tiene stock .

| CUXX - Nombre | CU05 - Compra |
| --- | --- |
| Descripción | Cliente desea tener un producto |
| Actores | Usuario |
| Precondición | El cliente y el producto debe poder ser identificado en el sistema y debe estar registrado. |
| Resumen | El cliente va a ver en la lista de de productos y seleccionar si alguno le interesa, Si el cliente está registrado y hay stock del producto el cliente lo comprara de lo contrario no podrá comprarlo. |
| Postcondición | En el primer caso, el producto sera restado en el stock, y en el segundo caso el cliente recibirá un mensaje de error ya sea por no estar registrado o por falta de stock . |
| Observaciones | Ninguna. |

| CUXX - Nombre | CU06 - Modificacion de Producto |
| --- | --- |
| Descripción | Modificar datos de un producto |
| Actores | Usuario |
| Precondición | El producto debe ser identificado en el sistema. |
| Resumen | Un producto identificado por id debe estar registrado en el sistema en ese cado será modificado por valores pasado por parámetro o en el caso contrario quedará todo igual que antes. |
| Postcondición | En el primer caso, el producto seleccionado será modificado, y en el segundo caso el cliente recibirá un mensaje de error que no se registra el producto . |
| Observaciones | Ninguna. |

User Story 05:

-Como usuario quiero poder comprar un producto en la lista de productos con stock.

-Como cliente quiero que el producto que selecciones para comprar tenga stock sino que no aparezca .

User Story 06:

-Como cliente quiero poder modificar el producto que quiera por completo.

-Como cliente quiero poder modificar solo un dato de un producto especifico

| CUXX - Nombre | CU07 - borrar Producto |
| --- | --- |
| Descripción | Elimina un producto |
| Actores | Usuario |
| Precondición | El producto debe estar registrado en el sistema. |
| Resumen | Debe verificar si el producto exista en ese caso se eliminará el producto con los datos o en el caso contrario mostrara un mensaje de error. |
| Postcondición | En el primer caso, el producto seleccionado será eliminado, y en el segundo caso el cliente recibirá un mensaje de error que no se registra el producto . |
| Observaciones | Ninguna. |

Como usuario quiero poder eliminar un producto en el sistema

Como usuario quiero que no se borre nada sin que se valide bien el producto a eliminar.

**Los Endpoints son:**

**Producto:**

**[GET]'/ '**

**encargado de traer todo los productos que tengan stock o dar un mensaje de error**

**[GET]'/:idProducto'**

**encargado de traer un producto especificado por id enviada por parámetro, de tipo “number”, o dar un mensaje de error**

**[PATCH]'/id' Recibe los datos del producto por body a cambiar y por parámetro su id, que será de tipo “number”, los modifica al producto que tenga ese id o da un mensaje de error. En caso de no recibir algún dato del producto, sólo actualizará los que reciba en el body.**

**[POST]'/comprar/:idproducto Recibe el id de producto por parametro, de tipo “number”,la cantidad y el método de pago por body. Compra un producto (resta la cantidad enviada en el body request) o da un mensaje de error. Aquí se aplica una regla de negocio: en caso de que el cliente envíe un método de pago el cual contenga un descuento (dictaminado por el contenido de la BBDD), se aplicará el mismo en el valor del artículo. También, en caso de que el cliente haga una compra mayor a 10 unidades, se le aplicará un descuento del 10%**

**[POST]'/' Recibe los datos del producto a agregar por el body .Valida la existencia de todos los campos que vienen en el body y crea un nuevo producto o da un mensaje de error.**

**[DELETE] '/id' Recibe un id de producto por parametro, si lo encuentra lo elimina o sino da un mensaje de error.**

**Categories:**

**[GET]'/ '**

**encargado de traer todo las categorías o dar un mensaje de error**

**[GET]'/idcategoria ' recibe un id**

**encargado de buscar una categoría con el id pasado por parámetro y listar todos sus productos o dar un mensaje de error**

**[POST]'/' Recibe los dato de la nueva categoria por el body ,los valida y lo agrega o da un mensaje de error**

**[PATCH] '/:id ' Recibe por parametro el id de categoria y en el body los datos de la categoria,lo que hace con el id es identificar la categoria a modificar y a esa categoria le cambia los datos por los pasados en el body. En caso de no recibir algún dato de la categoría, sólo actualizará los que reciba en el body.**

**[DELETE] '/:id ' Recibe por parametro el id,lo que hace con el id es identificar la categoria a eliminar y la borra de la base de datos o da un mensaje de error.**

**Usuario:**

**[GET]'/ '**

**encargado de traer todo los usuarios registrados o dar un mensaje de error**

**[GET]'/:id**

**encargado de traer un usuario especificado por id enviada por parametro o dar un mensaje de error**

**[POST]'/' Recibe los datos del usuario a agregar por el body .Valida la existencia de todos los campos que vienen en el body y crea un nuevo producto mientras no este repetido en el sistema o da un mensaje de error**

**[DELETE] '/id' Recibe un id de producto por parametro, si lo encuentra lo elimina o sino da un mensaje de error**

**[PATCH] '/id ' Recibe por parametro el id de usuario en el body los datos del usuario,lo que hace con el id es identificar al usuario validar los datos pasados en el body y lo modifica o da un mensaje de error**

**Index:**

**[GET]'/' Encagado de dar la bienvenida al usuario**

**[POST]'/login 'Recibe las credenciales por el body**

**Encargado de loguear al usuario por sus credenciales generando un token o dar un mensaje de error. Recibe en el cuerpo del request un valor login y un valor clave, ambos String.**