

Sprint 3 Datos y Backend

Duración: Semanas 5 y 6.
7 de noviembre / 20 de noviembre

Integrante: Manuel Julian Londoño Londoño

Grupo: 43

Proyecto: MyBisnes

Descripción:

MyBisnes es una aplicación web de compra y venta de productos de cualquier índole, esta aplicación permite que los usuarios se puedan registrar y ofrecer sus productos. Así mismo permitirá ver los productos de otras personas registradas en la aplicación y comprarlos.

Alcance:

Por el tiempo tan reducido para ejecutar el proyecto, se optará por realizar los siguientes módulos:

- Registrar usuario.
- Iniciar sesión en la aplicación.
- Registrar productos.
- Actualizar productos.
- Eliminar productos.
- Listar productos.

• Repositorio donde integrarán el trabajo.

<https://github.com/mjldondono/MyBisnes.git>

• Product Backlog.

Product Backlog				
Id	Nombre	Estado	Sprint	Tipo
1	Desarrollar Product Backlog	Hecho	1	Documentación
2	Configurar Repositorio en Github	Hecho	1	Configuración
3	Configurar entorno de desarrollo	Hecho	1	Configuración
4	Construir Historias de Usuario a desarrollar Sprint 1	Hecho	1	Documentación
5	Informe de Retrospectiva.	Hecho	1	Documentación
6	Sprint Backlog del siguiente Sprint.	Hecho	1	Documentación
7	Vídeo: Reunión Spring Review	Hecho	1	Documentación
8	Construir Archivo de evidencias Sprint 1	Hecho	1	Documentación
9	Actualizar Product Backlog	Hecho	2	Documentación
10	Historias desarrolladas en el Sprint 2	Hecho	2	Documentación
11	Vídeo: Reunión Spring Review	Hecho	2	Documentación
12	Historias desarrolladas en el Sprint 1.	Hecho	2	Documentación
13	Informe de Retrospectiva.	Hecho	2	Documentación
14	Diseño Base de Datos	Hecho	2	Desarrollo
15	Construir Frontend - Modulo Iniciar Sesión	Hecho	2	Desarrollo
16	Construir Frontend - Modulo Registrar Usuario	Hecho	2	Desarrollo
17	Construir Frontend - Modulo Registrar Producto	Hecho	2	Desarrollo
18	Sprint Backlog del siguiente Sprint.	Hecho	2	Documentación
19	Construir Archivo de evidencias Sprint 2	Hecho	2	Documentación
20	Actualizar Product Backlog	Hecho	3	Documentación
21	Historias desarrolladas en el Sprint 3.	Hecho	3	Documentación
22	Vídeo: Reunión Spring Review	Hecho	3	Documentación
23	Informe de Retrospectiva.	Hecho	3	Documentación
24	Sprint Backlog del siguiente Sprint.	Hecho	3	Documentación
25	Construir Backend - Modulo Iniciar Sesión	Hecho	3	Desarrollo
26	Construir Backend - Modulo Registrar Producto	Hecho	3	Desarrollo
27	Construir Backend - Modulo Actualizar Producto	Hecho	3	Desarrollo
28	Construir Backend - Modulo Eliminar Producto	Hecho	3	Desarrollo
29	Construir Backend - Modulo Listar Producto	Hecho	3	Desarrollo
30	Construir Archivo de evidencias Sprint 3	Hecho	3	Documentación
31	Construir Backend - Modulo Registrar Usuario	Sin Hacer	4	Desarrollo
32	Construir Frontend - Modulo Actualizar Producto	Sin Hacer	4	Desarrollo
33	Construir Frontend - Modulo Eliminar Producto	Sin Hacer	4	Desarrollo
34	Construir Frontend Modulo Listar Producto	Sin Hacer	4	Desarrollo
35	Historias desarrolladas en el Sprint 4	Sin Hacer	4	Documentación
36	Vídeo: Reunión Spring Review	Sin Hacer	4	Documentación
37	Informe de Retrospectiva.	Sin Hacer	4	Documentación
38	Sprint Backlog del siguiente Sprint. 5	Sin Hacer	4	Documentación
39	Despliegue de la aplicación.	Sin Hacer	4	Configuración

Backend - Modulo Iniciar Sesión

```

controllers > usuarios.controller.js > login > login > Usuario.findOne() callback
1  const Usuario = require("../models/usuarios.model");
2  const crypto = require("crypto");
3  const jwt = require("jsonwebtoken");
4
5  exports.login = function (req, res, next) {
6
7      let hashedpass = crypto.createHash("sha512").update(req.body.contrasena).digest("hex");
8
9      Usuario.findOne({ usuario: req.body.usuario, contrasena: hashedpass }, function (err, usuario) {
10
11          let response = {
12
13              token: null
14          }
15
16          if (usuario !== null) {
17              response.token = jwt.sign({
18                  id: usuario._id,
19                  usuario: usuario.usuario
20              }, "__recret__" //, { expiresIn: '12h' }
21          )
22          res.json(response);
23      }
24  }
25  })
26  }

```

Backend - Modulo Registrar Usuario

```

controllers > comerciantes.controller.js > create > create > comerciante.save() callback
1  const Comerciante = require("../models/comerciantes.model");
2
3  let response = {
4
5      msg: "",
6      exito: false
7  }
8
9
10 exports.create = function(req,res){
11
12     let comerciante = new Comerciante({
13
14         nombres: req.body.nombres,
15         apellidos: req.body.apellidos,
16         cedula: req.body.cedula,
17         telefono: req.body.telefono,
18         correo: req.body.correo,
19         direccion: req.body.direccion,
20         pais: req.body.pais,
21         departamento: req.body.departamento,
22         ciudad: req.body.ciudad
23     })
24 }

```

```
26   comerciante.save(function(err){
27
28       if(err){
29
30           console.error(err);
31           response.exito = false,
32           response.msg = "ERROR al GUARDAR el USUARIO"
33           res.json(response)
34           return;
35       }
36
37       response.exito = true;
38       response.msg = "El USUARIO se GUARDO CORRECTAMENTE"
39       res.json(response)
40
41   })
42 }
43
```

Backend - Modulo Registrar Producto

```
controllers > productos.controller.js > find > find
1  const Producto = require("../models/productos.model");
2
3  let response = {
4
5      msg: "",
6      exito: false
7  }
8  }
9
10 exports.create = function(req,res){
11
12     let producto = new Producto({
13
14         id_producto: req.body.id_producto,
15         nombre: req.body.nombre,
16         descripcion: req.body.descripcion,
17         marca: req.body.marca,
18         valor: req.body.valor
19     })
20 }
21
```

```
22     producto.save(function(err){
23
24         if(err){
25
26             console.error(err);
27             response.exito = false,
28             response.msg = "ERROR al GUARDAR el PRODUCTO"
29             res.json(response)
30             return;
31         }
32
33         response.exito = true;
34         response.msg = "El PRODUCTO se GUARDO CORRECTAMENTE"
35         res.json(response)
36
37     })
38 }
```

Backend - Modulo Actualizar Producto

```
58 exports.update = function(req, res){
59
60     let producto = ({
61
62         id_producto: req.body.id_producto,
63         nombre: req.body.nombre,
64         descripcion: req.body.descripcion,
65         marca: req.body.marca,
66         valor: req.body.valor
67
68     })
69 }
```

```
70 Producto.findByIdAndUpdate(req.params.id,{ $set: producto}, function(err){
71
72     if(err){
73
74         console.error(err),
75         response.exito = false,
76         response.msg = "ERROR al MODIFICAR el PRODUCTO"
77         res.json(response)
78         return;
79     }
80
81     response.exito = true,
82     response.msg = "El PRODUCTO se ACTUALIZO CORRECTAMENTE"
83     res.json(response)
84 })
85 }
```

Backend - Modulo Eliminar Producto

```
87 exports.remove = function(req, res){
88
89     Producto.findByIdAndRemove({_id: req.params.id}, function(err){
90
91         if(err){
92
93             console.error(err),
94             response.exito = false,
95             response.msg = "ERROR al ELIMINAR el PRODUCTO"
96             res.json(response)
97             return;
98         }
99
100         response.exito = true,
101         response.msg = "El PRODUCTO se ELIMINO CORRECTAMENTE"
102         res.json(response)
103     })
104 }
```

Backend - Modulo Listar Producto

```
40 exports.find = function(req, res){
41
42     Producto.find(function(err, productos){
43
44         res.json(productos);
45     })
46 }
47
48
49 exports.findOne = function(req, res){
50
51     Producto.findOne({_id: req.params.id},function(err, producto){
52
53         res.json(producto);
54     })
55 }
56 }
```

• Historias desarrolladas en el Sprint 3.

Historia de Usuario		
ID: 5		Nombre: editar producto
Prioridad en Negocio: Alta		Iteración Asignada:
Rol: Yo como usuario empresa		
Funcionalidad: Necesito poder editar un producto.		
Criterio de Aceptación:		
<ul style="list-style-type: none"> • El usuario selecciona la opción editar producto, el sistema mostrará el formulario diligenciado con las características del elemento, el usuario edita el producto y envía la información, el sistema valida la información y la edita el producto. 		
<ul style="list-style-type: none"> • El usuario cancela la acción editar el producto. 		
<ul style="list-style-type: none"> • El sistema rechaza los datos diligenciados del usuario, porque no cumple con los requerimientos de validación del formulario. 		

Historia de Usuario		
ID: 6		Nombre: eliminar producto
Prioridad en Negocio: Alta		Iteración Asignada:
Rol: Yo como usuario administrador o usuario de gestión		
Funcionalidad: Necesito poder eliminar producto.		
Criterio de Aceptación:		
<ul style="list-style-type: none"> • El usuario selecciona la opción eliminar producto, el sistema mostrará el mensaje de confirmación y el usuario acepta este mensaje, el sistema elimina el producto. 		
<ul style="list-style-type: none"> • El usuario no acepta el mensaje se confirmación de eliminar. 		
<ul style="list-style-type: none"> • El usuario cancela la acción de eliminación del producto. 		

• Vídeo: Reunión Spring Review.

<https://youtu.be/R5ku7h2MjAs>

• Informe de Retrospectiva.

Definida el diseño, se avanzó bastante en el backend, ahora iniciamos con el frontend en los otros módulos faltante.

Buscamos otras asesorías a la asignada y continuamos avanzado en el proyecto.

Debemos aumentar las horas dedicación al proyecto para aumentar el tiempo en el diseño y codificación.

La desmotivación predomina y se busca una forma de continuar el proyecto.

• Sprint Backlog del siguiente Sprint.

Product Backlog				
Id	Nombre	Estado	Sprint	Tipo
1	Construir Backend - Modulo Registrar Usuario	Sin Hacer	4	Desarrollo
2	Construir Frontend - Modulo Actualizar Producto	Sin Hacer	4	Desarrollo
3	Construir Frontend - Modulo Eliminar Producto	Sin Hacer	4	Desarrollo
4	Construir Frontend Modulo Listar Producto	Sin Hacer	4	Desarrollo
5	Historias desarrolladas en el Sprint 4	Sin Hacer	4	Documentación
6	Vídeo: Reunión Spring Review	Sin Hacer	4	Documentación
7	Informe de Retrospectiva.	Sin Hacer	4	Documentación
8	Sprint Backlog del siguiente Sprint. 5	Sin Hacer	4	Documentación
9	Despliegue de la aplicación.	Sin Hacer	4	Configuración