

Reporte de Inspección				
Fecha de Inicio:		22/10/2018		
Revisores:	Nombre		Rol	
	Garay Lucio Omar		Revisor/Moderador	
	Pecorari Zárate Franco Misael		Inspector	
	Oscares Álvaro		Lector	
	Chávez Flores Alexa Ceciclia		Autor	
Alcance de la revisión:		Porción de código de clase PedidoLoqueseaPage de la línea 1 a la 226		
Nombre		Versión	Autor	
Clase PedidoLoqueseaPage		Inicial	@Grupo5	
Hallazgos				
Nº	Fecha	Descripción	Línea	Severidad
1	22/10/2018	Import ElementRef no utilizado	1	Menor
2	22/10/2018	Import FormControl no utilizado	7	Menor
3	22/10/2018	Import GoogleMapsEvent no utilizado	12	Menor
4	22/10/2018	Import GoogleMapOptions no utilizado	13	Menor
5	22/10/2018	Import CameraPosition no utilizado	14	Menor
6	22/10/2018	Import MarkerOptions no utilizado	15	Menor
7	22/10/2018	Import Marker no utilizado	16	Menor
8	22/10/2018	No sigue las reglas de importado definidas en la hoja de estilo	1 - 22	Menor
9	22/10/2018	Nombre de clase no respeta CamelCase ni reglas de nombrado	34	Menor
10	22/10/2018	Nombre de variable “plt” no representativo	54	Menor
12	22/10/2018	Espacio entre if y los paréntesis	114, 124, 130, 156	Menor
13	22/10/2018	Nombre de método no representativo	106	Menor
14	22/10/2018	Nombre de variable mal escrito “opciones”	195	Cosmético
Decisión Final:		Acepar Provisoriamente		

Código inspeccionado

```
1. import { Component, ElementRef } from '@angular/core';
2. import { IonicPage, NavController, NavParams, ToastController, Platform } from 'ionic-
  angular';
3. import { ViewChild } from '@angular/core';
4. import { Slides } from 'ionic-angular';
5. import { Producto } from '../interfaces/producto.interface';
6. import { PedidoLoQueSea } from '../interfaces/pedido-loquesea.interface';
7. import { FormBuilder, FormControl, FormGroup, Validators } from '@angular/forms';
8. import { Camera, CameraOptions } from '@ionic-native/camera';
9. import {
10.   GoogleMaps,
11.   GoogleMap,
12.   GoogleMapsEvent,
13.   GoogleMapOptions,
14.   CameraPosition,
15.   MarkerOptions,
16.   Marker,
17.   LatLng
18. } from '@ionic-native/google-maps';
19. import { ImagePicker, ImagePickerOptions } from '@ionic-native/image-picker';
20. import { FormasDePago } from '../app/app.const';
21. import { Coordenadas } from '../interfaces/coordenadas.interface';
22. import { Geolocation } from '@ionic-native/geolocation'
23. /**
24.  *Generated class for the PedidoLoqueseaPage page.
25.  *
26.  *See https://ionicframework.com/docs/components/#navigation for more info on
27.  *Ionic pages and navigation.
28.  */
29. @IonicPage()
30. @Component({
31.   selector: 'page-pedido-loquesea',
32.   templateUrl: 'pedido-loquesea.html',
33. })
34. export class PedidoLoqueseaPage {
35.   FormasDePago = FormasDePago;
36.   @ViewChild(Slides) slides: Slides;
37.   private map:GoogleMap;
38.   private location:LatLng = new LatLng(0,0);
39.   //Formularios
40.   pasoUno: FormGroup;
41.   pasoDos: FormGroup;
42.   pasoTres: FormGroup;
43.   fechaActivada:boolean=false;
44.   //Variables
45.   producto: Producto = { direccionComercio: undefined, descripcion: undefined, foto:
     "assets/img/sin_imagen.jpg" };
46.   pedido: PedidoLoQueSea;
47.   direccionPorTexto: boolean = true;
48.   constructor(public navCtrl: NavController,
49.     public NavParams: NavParams,
```

```
50. public formBuilder: FormBuilder,
51. public camera: Camera,
52. public toastCtrl: ToastController,
53. public geolocation: Geolocation,
54. public plt: Platform,
55. private googleMaps: GoogleMaps, public imagePicker: ImagePicker) {
56. //Iniciando variables
57. this.inicializarPedido();
58. this.inicializarProducto();
59. //Formularios
60. this.pasoUno = this.formBuilder.group({
61. descripcion: ['', Validators.required],
62. foto: [''],
63. direccionComercio: [''],
64. coordenadas: ['']
65. });
66. this.pasoDos = this.formBuilder.group({
67. domicilio: ['', Validators.required],
68. fechaEntrega: ['', Validators.required]
69. });
70. this.construirFormularioPaso3("FormasDePago.Contado");
71. }
72. switchFecha(){
73. this.fechaActivada = !this.fechaActivada;
74. if(!this.fechaActivada){
75. this.pedido.fechaEntrega = new Date().toLocaleString();
76. }
77. else{
78. this.pedido.fechaEntrega = undefined;
79. }
80. }
81. construirFormularioPaso3(formaDePago: string) {
82. console.log(formaDePago);
83. console.log(this.pedido);
84. let esContado = formaDePago === "FormasDePago.Contado";
85. this.pasoTres = this.formBuilder.group({
86. formaDePago: ['', Validators.required],
87. pagaCon: ['', esContado ? Validators.required : undefined],
88. nroTarjeta: ['', Validators.compose( esContado? [] :[Validators.required,
      Validators.maxLength(16), Validators.minLength(16)])],
89. titular: ['', esContado? [] :Validators.required],
90. vencimiento: ['', Validators.compose(esContado? [] : [Validators.required,
      Validators.pattern("[0-9][0-9]-[0-9][0-9]"))],
91. codigoSeguridad: ['', Validators.compose(esContado? [] :[Validators.required,
      Validators.maxLength(3), Validators.minLength(3)])]
92. })
93. }
94. inicializarProducto() {
95. this.producto = { direccionComercio: undefined, descripcion: undefined, foto: undefined };
96. }
97. inicializarPedido() {
```

```
98. this.pedido = { productos: [], domicilio: undefined, formaDePago: FormasDePago.Contado,
    fechaEntrega: new Date().toISOString() };
99. }
100.     bloquearSlides() {
101.         this.slides.lockSwipes(true)
102.     }
103.     desbloquearSlides() {
104.         this.slides.lockSwipes(false)
105.     }
106.     ionViewDidLoad() {
107.         this.bloquearSlides();
108.         this.slides.centeredSlides = false;
109.     }
110.     validarPaso(paso: number): boolean {
111.         let res: boolean = true;
112.         switch (paso) {
113.             case 0:
114.                 if (!this.pedido.productos.length) {
115.                     let alert = this.toastCtrl.create({
116.                         message: "¡Agregá al menos un producto!",
117.                         duration: 1500
118.                     });
119.                     alert.present();
120.                     res = false;
121.                 }
122.                 break;
123.             case 1:
124.                 if (!this.pasoDos.valid) {
125.                     res = false;
126.                     this.mostrarAlertaCamposRequeridos();
127.                 }
128.                 break;
129.             case 2:
130.                 if (!this.pasoTres.valid) {
131.                     res = false;
132.                     this.mostrarAlertaCamposRequeridos();
133.                 }
134.                 break;
135.             }
136.         return res;
137.     }
138.     mostrarAlertaCamposRequeridos(){
139.         const alert = this.toastCtrl.create({
140.             message: "¡Completa todos los campos requeridos (*) antes de continuar!",
141.             duration: 1500
142.         });
143.         return alert.present();
144.     }
145.     mostrarExito(){
146.         const alert = this.toastCtrl.create({
147.             message: "¡Se ha registrado el pedido!",
148.             closeButtonText: '¡Genial!',
```

```
149.     showCloseButton: true
150.   });
151.   return alert.present();
152. }
153. //Navega entre los pasos del slide, si el parametro "avanzar" es true (por
154. defecto lo es), avanza hacia la derecha, si no, hacia la izquierda.
155. navegar(avanzar: boolean = true) {
156.   if (this.validarPaso(this.slides.getActiveIndex()) || !avanzar) {
157.     this.desbloquearSlides();
158.     avanzar ? this.slides.slideNext(200) : this.slides.slidePrev(200);
159.     this.bloquearSlides();
160.   }
161. }
162. listo() {
163.   if(this.validarPaso(2)){
164.     this.mostrarExito().then(()=>{
165.       this.navCtrl.popToRoot();
166.     })
167.   }
168. }
169. esUltimoPaso(): boolean {
170.   return this.slides.isEnd();
171. }
172. esPrimerPaso(): boolean {
173.   return this.slides.isBeginning();
174. }
175. tomarFoto() {
176.   const options: CameraOptions = {
177.     quality: 60,
178.     destinationType: this.camera.DestinationType.DATA_URL,
179.     encodingType: this.camera.EncodingType.JPEG,
180.     sourceType: this.camera.PictureSourceType.PHOTOLIBRARY,
181.     mediaType: this.camera.MediaType.PICTURE
182.   }
183.   this.camera.getPicture(options).then((imageData) => {
184.     let base64Image = 'data:image/jpeg;base64,' + imageData;
185.     this.producto.foto = base64Image;
186.   }, (err) => {
187.     alert(err);
188.   });
189. }
190. mostrarMapa() {
191.   this.direccionPorTexto = false;
192.   this.navCtrl.push('MapaPage', {producto: this.producto});
193. }
194. seleccionarFoto(){
195.   let opciones: ImagePickerOptions = {
196.     quality: 50,
197.     maximumImagesCount: 1,
198.     outputType: 1,
199.   }
200.   this.imagePicker.getPictures(opciones).then((foto)=>{
```

```
201.     this.producto.foto = foto[0];
202.     })
203.     }
204.     eliminarCoordenadas() {
205.     this.direccionPorTexto = true;
206.     this.producto.direccionComercio = undefined;
207.     }
208.     agregarProducto() {
209.     if(!this.producto.direccionComercio){
210.     this.mostrarAlertaCamposRequeridos();
211.     }
212.     else{
213.     this.pedido.productos.push(this.producto);
214.     this.inicializarProducto();
215.     this.pasoUno.reset();
216.     const agregadoACarrito = this.toastCtrl.create({
217.     showCloseButton: true,
218.     closeButtonText: "OK",
219.     message: "¡Se ha agregado al carrito!",
220.     dismissOnPageChange: true
221.     })
222.     agregadoACarrito.present();
223.     console.log(this.pedido);
224.     }
225.     }
226.     }
```