

DOCUMENTACION PROYECTO

PET MASTER

EQUIPO #4

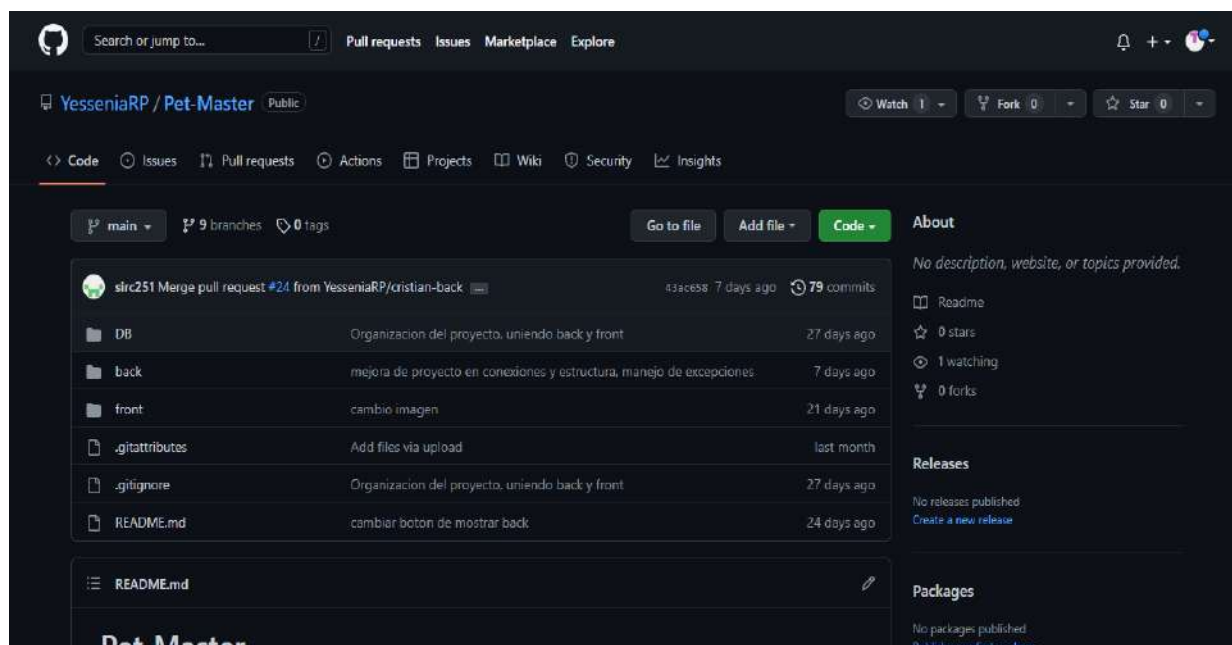
INTEGRANTES:

Cristian Ricardo Ramírez Rojas
Isaías Daniel Castilla Reyes
Diego Rincón Triana
Carlos Andrés Vinasco Soto
Yesenia María Rosero Portillo

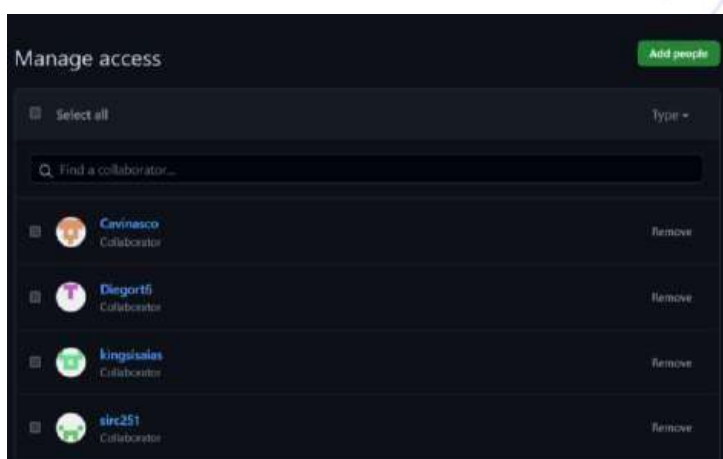
Descripción de roles del equipo.

Rol	Integrante	Descripción	Tareas
Gestor de base de datos	Diego Rincón Triana	Encargado de gestionar base de datos	Crear base de datos
Líder de equipo	Yesenia María Rosero Portillo	Encargado de entrenar al equipo	Apoyar en la planificación y desarrollo de cada Sprint
Desarrollador Backend	Cristian Ricardo Ramírez Rojas	Encargado del desarrollo del Software	Desarrollar Backend
Desarrollador Frontend	Isaías Daniel Castilla Reyes	Encargado del desarrollo del Software	Desarrollar Frontend
Tester	Carlos Andrés Vinasco Soto	Encargado de realizar las pruebas de funcionalidad	Planificar y llevar a cabo pruebas de software de los ordenadores para comprobar si funcionan correctamente

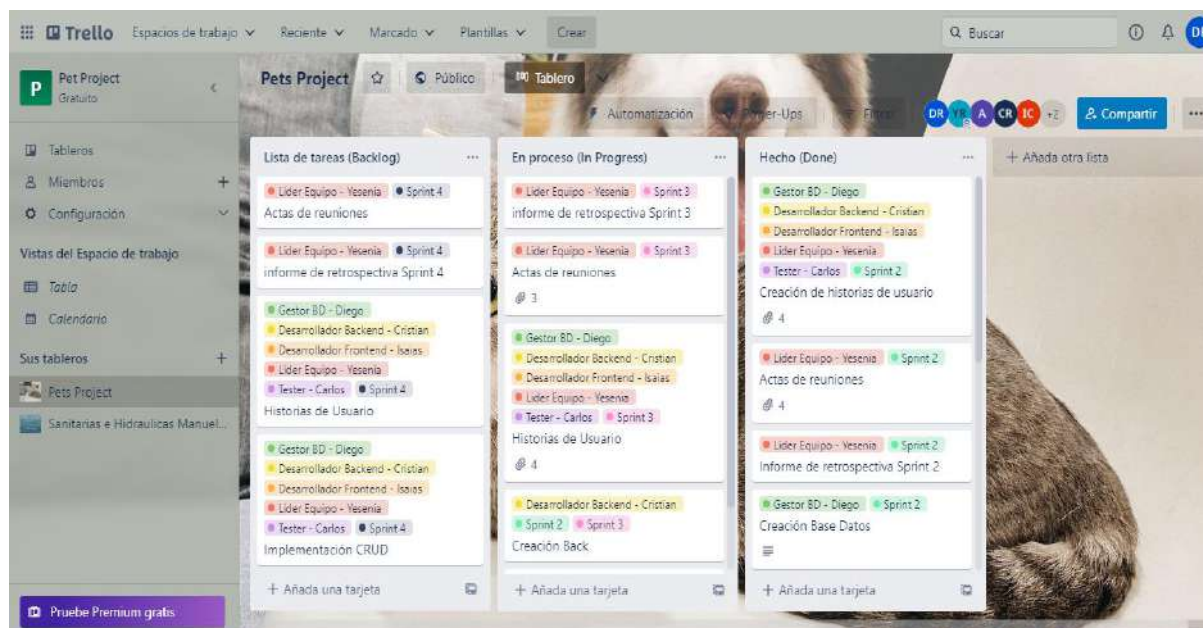
Repositorio (GitHub) donde integrarán el trabajo.



<https://github.com/YesseniaRP/Pet-Master>

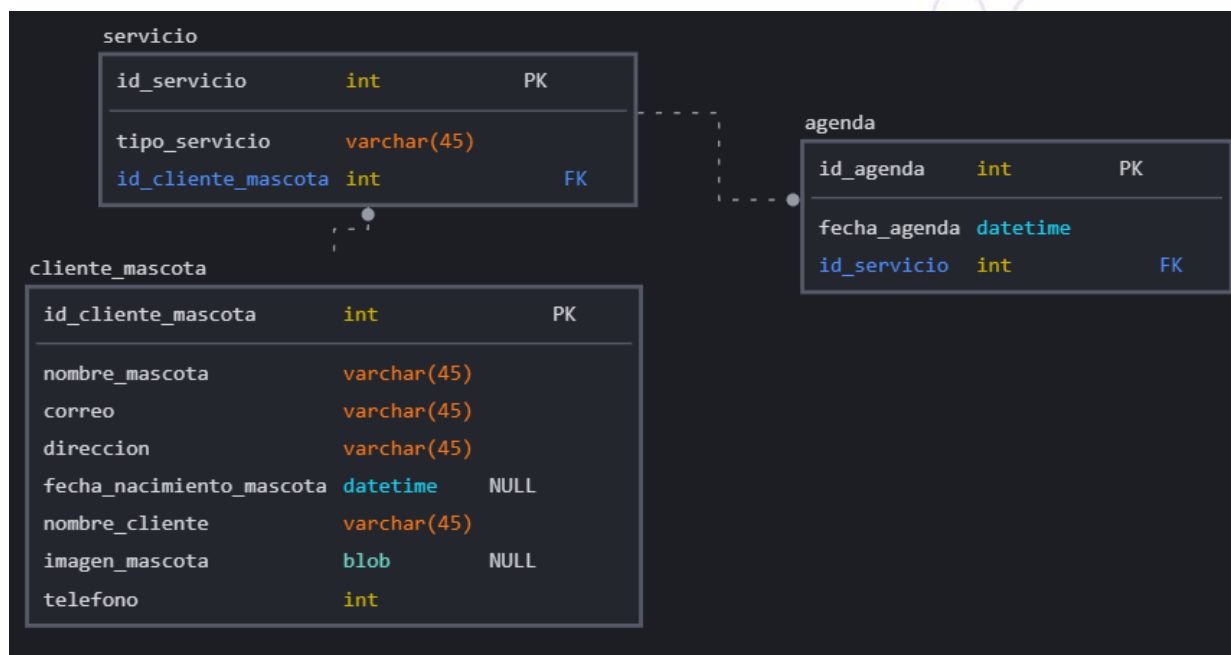


Trello.



<https://trello.com/b/RwQvkT0w/pets-project>

Diagrama Entidad Relación.



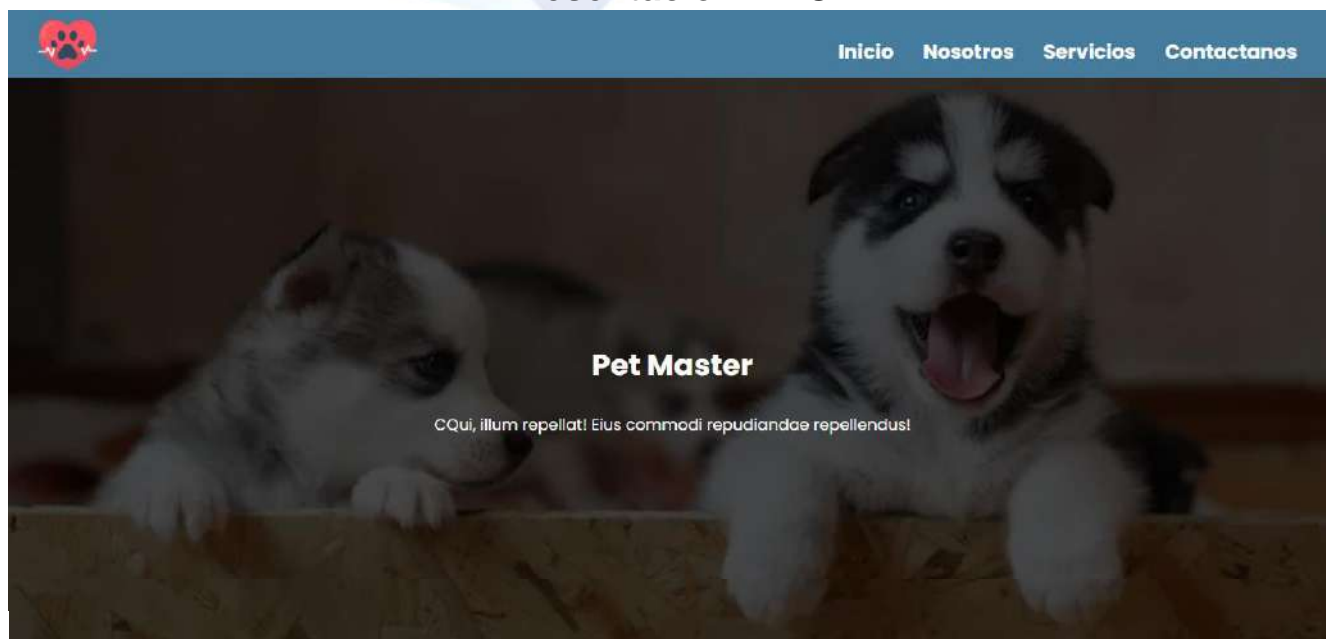
Script de la Base de Datos con Schema y Data.

```
CREATE TABLE `cliente_mascota` (
  `id_cliente_mascota` INT(11) NOT NULL,
  `nombre_mascota` VARCHAR(45) NOT NULL,
  `correo` VARCHAR(45) NOT NULL,
  `direccion` VARCHAR(45) NOT NULL COLLATE,
  `fecha_nacimiento_mascota` DATE NULL DEFAULT NULL,
  `nombre_cliente` VARCHAR(45) NOT NULL,
  `imagen_mascota` BLOB NULL DEFAULT NULL,
  `telefono` VARCHAR(45) NOT NULL DEFAULT,
  PRIMARY KEY (`id_cliente_mascota`);
```

```
CREATE TABLE `agenda` (
  `id_agenda` INT(11) NOT NULL,
  `id_servicio` INT(11) NOT NULL,
  `fecha_agenda` DATETIME NOT NULL,
  PRIMARY KEY (`id_agenda`),
  INDEX `FK_servicio` (`id_servicio`),
  CONSTRAINT `FK_servicio` FOREIGN KEY (`id_servicio`) REFERENCES
`petmaster`.`servicio` (`id_servicio`);
```

```
CREATE TABLE `servicio` (
  `id_servicio` INT(11) NOT NULL,
  `id_cliente_mascota` INT(11) NOT NULL,
  `tipo_servicio` VARCHAR(45) NOT NULL,
  PRIMARY KEY (`id_servicio`),
  INDEX `FK_cliente_mascota` (`id_cliente_mascota`),
  CONSTRAINT `FK_cliente_mascota` FOREIGN KEY (`id_cliente_mascota`)
REFERENCES `petmaster`.`cliente_mascota` (`id_cliente_mascota`);
```


Presentación MVC.



Tenemos estas opciones para tí



MÁS SERVICIOS →

Mascotas de nuestros clientes



Todos los derechos reservados by Pet-Master ©

Navegación entre interfases.

Opción Botón Nosotros.



Sobre nosotros



Lorem ipsum dolor sit amet, consectetur adipisicing elit. Omnis, rerum quaeat a magni illo dolor autem suscipit reiciendis doloribus odit dicta vitae consequuntur exercitationem eligendi quod harum libero repellendus id? Lorem ipsum dolor sit amet, consectetur adipisicing elit. Omnis, rerum quaeat a magni illo dolor autem suscipit reiciendis doloribus odit dicta vitae consequuntur exercitationem eligendi quod harum libero repellendus id?

Lorem ipsum dolor sit amet, consectetur adipisicing elit. Omnis, rerum quaeat a magni illo dolor autem suscipit reiciendis doloribus odit dicta vitae consequuntur exercitationem eligendi quod harum libero repellendus id? Lorem ipsum dolor sit amet, consectetur adipisicing elit. Omnis, rerum quaeat a magni illo dolor autem suscipit reiciendis doloribus odit dicta vitae consequuntur exercitationem eligendi quod harum libero repellendus id?

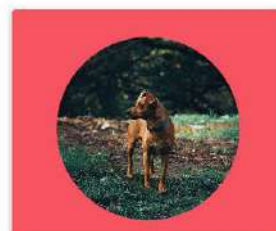


Todos los derechos reservados by Pet-Master ©

Opción Botón Servicios.



Tenemos estas opciones para ti



Todos los derechos reservados by Pet-Master ©

Opción Botón Servicios Banner Clínica.

[Inicio](#)
[Nosotros](#)
[Servicios](#)
[Contactanos](#)

Tenemos estas opciones para ti

Clínica

Lorem ipsum dolor sit amet
consectetur adipiscing elit.
Minimo, in?

AGENDAR CITA

Todos los derechos reservados by Pet-Master ©

Opción Botón Servicios Banner Clínica al dar Clic en "Agendar cita" se despliega un formulario, esta opción es igual para cada servicio (siguientes banners).



Pet Master

Ingrese los siguientes datos

Información personal

NOMBRE

Ingrese su nombre

E-MAIL

Ingrese su E-mail

TELÉFONO

Ingrese su teléfono

Información de su mascota

NOMBRE

Ingrese el nombre de su mascota

TIPO DE MASCOTA

-- Seleccione --

Información del servicio

ESCOJA EL SERVICIO

-- Seleccione --

FECHA:

dd/mm/aaaa

HORA:

--:--:--

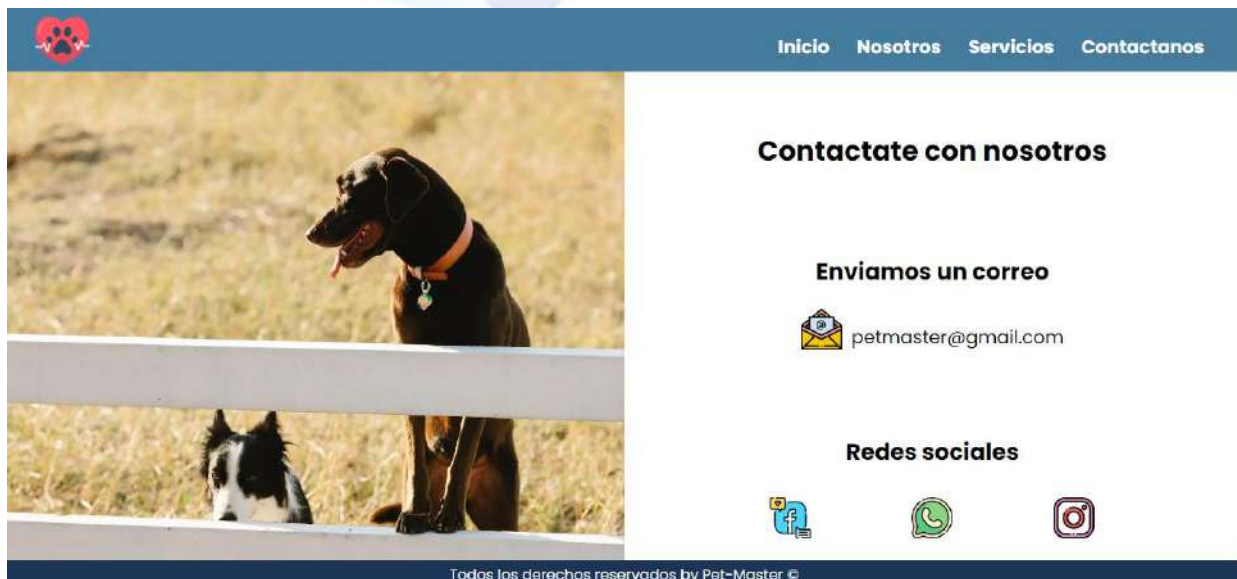
MENSAJE

ENVIAR

Todos los derechos reservados by Pet-Master ©

Opción Botón Contáctanos.

Opción Botón Contáctanos.



Aplicación con Persistencia Relacional.

ENTIDAD CLIENTE_MASCOTA

```
19 usages Cristian Ramirez
@Entity(name= "cliente_mascota")
@Builder
@NoArgsConstructor
@AllArgsConstructor
@Setter
@Getter
public class ClienteMascota {

    @Id
    private int idClienteMascota;
    @Column
    private String nombreMascota;
    @Column
    private String correo;
    @Column
    private String direccion;
    @Column(columnDefinition = "DATE")
    @JsonFormat(shape = JsonFormat.Shape.STRING, pattern = "yyyy-MM-dd")
    private Date fechaNacimientoMascota;
    @Column
    private String nombreCliente;
    @Column
    private byte[] imagenMascota;
    @Column
    private int telefono;
    @Column
    private String descripcion;
```

ENTIDAD AGENDA

```
import lombok.*;

import javax.persistence.*;
import java.sql.Date;

7 usages  Cristian Ramirez
@Entity
@Table(name= "agenda")
@Builder
@NoArgsConstructor
@AllArgsConstructor
@Setter
@Getter
public class Agenda {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private int idAgenda;
    @Column(name = "id_servicio")
    private int idServicio;
    @Column(name="id_cliente_mascota")
    private int idClienteMascota;
    @Column(name="fecha_agenda")
    private Date fechaAgenda;
}
```

SERVICIO CON LISTAR, CREAR Y BORRAR CLIENTE_MASCOTA

[illegible]

SERVICIO AGENDAR CITA

```
1 package com.project.projectpet.service.impl;
2
3 import com.fasterxml.jackson.databind.ObjectMapper;
4 import com.project.projectpet.controller.dto.AgendarCitaDTO;
5 import com.project.projectpet.entity.Agenda;
6 import com.project.projectpet.repository.AgendaRepository;
7 import com.project.projectpet.service.AgendarCitaService;
8 import org.springframework.beans.factory.annotation.Autowired;
9 import org.springframework.stereotype.Repository;
10 import org.springframework.stereotype.Service;
11
12 import javax.transaction.Transactional;
13
14 @Service
15 @Transactional
16 public class AgendarCitaServiceImpl implements AgendarCitaService {
17     1 usage
18     @Autowired
19     private AgendaRepository agendaRepository;
20
21     1 usage
22     @Override
23     public AgendarCitaDTO save(AgendarCitaDTO agendarCita) {
24         ObjectMapper mapper = new ObjectMapper();
25         Agenda agenda = mapper.convertValue(agendarCita, Agenda.class);
26         return mapper.convertValue(agendaRepository.save(agenda), AgendarCitaDTO.class);
27     }
28 }
```

REPOSITORIO CLIENTE_MASCOTA

```
1 package com.project.projectpet.repository;
2
3 import com.project.projectpet.entity.ClienteMascota;
4 import org.springframework.data.jpa.repository.JpaRepository;
5 import org.springframework.stereotype.Repository;
6
7 @Repository
8 public interface ClienteMascotaRepository extends JpaRepository<ClienteMascota, Integer> {
9
10 }
11 }
```


REPOSITORIO AGENDA

```
package com.project.projectpet.repository;

import ...

2 usages Cristian Ramirez
@Repository
public interface AgendaRepository extends JpaRepository<Agenda, Integer> {

}
|
```

LISTAR CONTROLADOR CLIENTE_MASCOTA

```
ClienteMascotaRepository.java x AgendaRepository.java x ClienteMascotaController.java x ClienteMascota.java x ClienteMascotaService.java x
import java.util.Optional;

Cristian Ramirez
@RestController
@Component
@RequestMapping("/v1/clientemascota")
public class ClienteMascotaController {

4 usages
@Autowired
private ClienteMascotaService clienteMascotaService;

1 usage
@Autowired
private AgendarCitaService agendarCitaService;

Cristian Ramirez
@Operation(summary = "Obtiene todos los datos de la tabla ClienteMascota")
@ApiResponses(value = {
    @ApiResponse(responseCode = "200", description = "Encontro los datos",
        content = {@Content(mediaType = "application/json",
            schema = @Schema(implementation = ClienteMascota.class))}),
    @ApiResponse(responseCode = "500", description = "Error interno",
        content = @Content)})
@GetMapping("/allClientesMascotas")
ResponseEntity<List<ClienteMascota>> all() {
    return new ResponseEntity<List<ClienteMascota>>(clienteMascotaService.findAll(), headers, HttpStatus.OK);
}
```

CONTROLADOR CREAR, ELIMINAR CLIENTE_MASCOTA

```

ClienteMascotaController
ClienteMascotaControllerExceptionHandler
entity
  Agenda
  ClienteMascota
  Servicio
exceptions
repository
  AgendaRepository
  ClienteMascotaRepository
  ServicioRepository
service
  AgendarCitaServiceImpl
  ClienteMascotaServiceImpl
  AgendarCitaService
  ClienteMascotaService

66
67 @DeleteMapping("/{eliminar/{id}}")
68 ResponseEntity<HttpStatus> delete(@PathVariable("id") Integer id) {
69     clienteMascotaService.delete(id);
70     return new ResponseEntity<>(HttpStatus.NO_CONTENT);
71 }
72
73
74 @PostMapping("/{crear}")
75 public ClienteMascota guardar(@RequestBody @Valid ClienteMascota clienteMascota) {
76     return clienteMascotaService.create(clienteMascota);
77 }
78

```


CONTROLADOR AGENDA

```
package com.project.projectpet.controller.dto;

import lombok.*;

import javax.validation.constraints.NotNull;
import java.io.Serializable;
import java.sql.Date;

11 usages  ± Cristian Ramirez
@Getter
@Setter
@Builder(toBuilder = true)
@AllArgsConstructor
@NoArgsConstructor
public class AgendarCitaDTO implements Serializable {
    Integer idAgenda;
    Integer idServicio;
    Integer idClienteMascota;
    Date fechaAgenda;
}
```

Pruebas Unitarias de la Lógica Desarrollada Postman.

CONTROLADOR AGENDA

POST http://localhost:8080/v1/clientemascota/Crear

Params Authorization Headers (8) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
1 {
2   "idClienteMascota": 3,
3   "nombreMascota": "Roberto",
4   "correo": "camilo@lorena.com",
5   "direccion": "Calle 2",
6   "fechaNacimientoMascota": "1019-12-12",
7   "nombreCliente": "Erick",
8   "imagenMascota": null,
9   "telefono": 3373937,
10  "descripcion": "Pruebacrear"
11 }
```

Body Cookies Headers (5) Test Results

Status: 200 OK Time: 471 ms Size: 393 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "idClienteMascota": 3,
3   "nombreMascota": "Roberto",
4   "correo": "camilo@lorena.com",
5   "direccion": "Calle 2",
6   "fechaNacimientoMascota": "1019-12-12",
7   "nombreCliente": "Erick",
8   "imagenMascota": null,
9   "telefono": 3373937,
10  "descripcion": "Pruebacrear"
11 }
```

Activar Windows
Ve a Configuración para activar Windows.

Cookies Capture requests Bootcamp Runner Trash

PRUEBA LISTAR TABLA PRINCIPAL CLIENTE_MASCOTA

GET http://localhost:8080/v1/clientemascota/allClientesMascotas Send

Params Authorization Headers (5) Body Pre-request Script Tests Settings Cookies

Query Params

KEY	VALUE	DESCRIPTION	Bulk Edit

Body Cookies Headers (5) Test Results Status: 200 OK Time: 548 ms Size: 837 B Save Response

Pretty Raw Preview Visualize JSON Copy Search

```

1 {
2   "idClienteMascota": 1,
3   "nombreMascota": "toby",
4   "correo": "toby@gmail.com",
5   "direccion": "dirección",
6   "fechaNacimientoMascota": "2022-01-01",
7   "nombreCliente": "Maru",
8   "imagenMascota": null,
9   "telefono": 123123123,
10  "descripcion": "Nada"
11 }
12 ,
13 {
14   "idClienteMascota": 2,
15   "nombreMascota": "Tony",
16   "correo": "Tony@gmail.com",
17   "direccion": "sdasd",
18   "fechaNacimientoMascota": "2020-01-01",
19   "nombreCliente": "Cristian",
20   "imagenMascota": null,
21   "telefono": 3373937,
22   "descripcion": "Prueba1"
23 }
24 ]
  
```

Activar Windows
Ve a Configuración para activar Windows

PRUEBA ELIMINAR TABLA PRINCIPAL CLIENTE_MASCOTA

New Collection / **eliminar(id)** Save ... Copy Search

DELETE http://localhost:8080/v1/clientemascota/eliminar/3 Send

Params Authorization Headers (6) Body Pre-request Script Tests Settings Cookies

Query Params

KEY	VALUE	DESCRIPTION	Bulk Edit
Key	Value	Description	

Body Cookies Headers (3) Test Results Status: 204 No Content Time: 564 ms Size: 112 B Save Response

Pretty Raw Preview Visualize Text Copy Search

```

1 |
  
```

ENTIDAD GUARDAR TABLA (ASIGNAR CITA)

POST http://localhost:8080/v1/clientemascota/agendar-cita Send

Params Authorization Headers (8) **Body** Pre-request Script Tests Settings Cookies Beautify

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL ☒ JSON

```

1  {
2    "idAgenda": null,
3    "idServicio": 1,
4    "idClienteMascota": 1,
5    "fechaAgenda": "2022-01-01"
6  }

```

Body Cookies Headers (5) Test Results Status: 200 OK Time: 700 ms Size: 242 B Save Response

Pretty Raw Preview Visualize **JSON**

```

1  {
2    "idAgenda": 10,
3    "idServicio": 1,
4    "idClienteMascota": 1,
5    "fechaAgenda": "2021-12-31"
6  }

```

New Collection / **/asignarCitas** Save ...

POST http://localhost:8080/v1/clientemascota/Crear Send

Params Authorization Headers (8) **Body** Pre-request Script Tests Settings Cookies Beautify

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL ☒ JSON

```

6    "fechaNacimientoMascota": "1019-12-12",
7    "nombreCliente": "Erick",
8    "imagenMascota": null,
9    "telefono": 3373937,
10   "descripcion": "Pruebacrear"
11  }

```

Body Cookies Headers (5) Test Results Status: 200 OK Time: 471 ms Size: 303 B Save Response

Pretty Raw Preview Visualize **JSON**

```

1  {
2    "idClienteMascota": 3,
3    "nombreMascota": "Roberto",
4    "correo": "camilo@lorena.com",
5    "direccion": "Calle 2",
6    "fechaNacimientoMascota": "1019-12-12",
7    "nombreCliente": "Erick",
8    "imagenMascota": null,
9    "telefono": 3373937,
10   "descripcion": "Pruebacrear"
11  }

```

Activar Windows
Ve a Configuración para activar Windows.

Cookies Capture requests Bootcamp Runner Trash

Historias de Usuario Desarrolladas en el Sprint 3.

<https://trello.com/b/RwQvkT0w/pets-project>

Historia de Usuario	
ID:1	Nombre: Calendario
Prioridad en Negocio:	Iteración Asignada:
Rol: Yo como usuario final	
Funcionalidad: Solicito un calendario para escoger fecha	
Criterio de Aceptación:	
<ul style="list-style-type: none"> El usuario accede a la página web y en la opción servicios selecciona a través de un calendario la fecha y hora para apartar su cita 	

Historia de Usuario	
ID:1	Nombre: Control citas
Prioridad en Negocio:	Iteración Asignada:
Rol: Administrador	
Funcionalidad: Quiero ver las citas agendadas	
Criterio de Aceptación:	
<ul style="list-style-type: none"> El administrador accede a la plataforma para ver las citas agendadas 	

Historia de Usuario	
ID:1	Nombre: Pruebas de persistencia
Prioridad en Negocio:	Iteración Asignada:
Rol: Administrador	
Funcionalidad: Requiere revisar persistencia relacional	
Criterio de Aceptación:	
<ul style="list-style-type: none"> El Back-end inicia pruebas de solicitudes desde Front al Back y quede persistencia en la DB 	

Historia de Usuario	
ID:1	Nombre: Pruebas
Prioridad en Negocio:	Iteración Asignada:
Rol: Administrador	
Funcionalidad: Quiero ver las pruebas	
Criterio de Aceptación:	
<ul style="list-style-type: none"> El Back realiza pruebas con Postman para comprobar su funcionalidad 	

Informe de Retrospectiva Sprint 3.

¿Que salió bien en las iteraciones? (aciertos)	¿Qué no salió bien en las iteraciones? (errores)	¿Qué mejoras vamos a implementar en la próxima iteración? (mejora continua).
Manejo Trello, GitHub, Heroku	Nos hackearon el Heroku y no se pudo ingresar toco realizar otra sesion	Se organizo que solo sea una tutoría Scrum a la semana por practicidad y disponibilidad de tiempo.
El avance del proyecto en las tutorias.	Falta de claridad en los temas a desarrollar.	Actualizacion del Github y Trello.
Mejoramos nuestra organización del repositorio	Pocas reuniones de grupo para definir las correcciones del proyecto.	Avances con Back-End

Historias de Usuario a Desarrollas en el Sprint 4.

Historia de Usuario	
ID:1	Nombre: Despliegue Back-End
Prioridad en Negocio:	Iteración Asignada:
Rol: Administrador	
Funcionalidad: Requiero probar BackEnd	
Criterio de Aceptación:	
<ul style="list-style-type: none"> El Back-End realiza el despliegue en la nube. 	

Historia de Usuario	
ID:1	Nombre: Despliegue DB en local/nube
Prioridad en Negocio:	Iteración Asignada:
Rol: Administrador	
Funcionalidad: Requiero probar DB	
Criterio de Aceptación:	
<ul style="list-style-type: none"> El Back-End realiza el despliegue de la Data Base en la nube. 	