

PRACTICA DE PRUEBAS DE SOFTWARE

CASO: SISTEMA DE ATENCION VETERINARIA.

Para hacer el sistema tengo que identificar mis variables en este caso estoy tomando:

Registro paciente:

Agendar cita:

Agregar inventario:

Generar factura:

Para este caso se entiendo que registrar paciente es ingresar un paciente al sistema, para la agenda de cita es para citar al paciente a una cita para ver su condición, el agregar inventario es para los medicamentos que se esta usando para curar al paciente y por ultima la generación de la factura es para darle una boleta la paciente por los servicios prestados dentro de la veterinaria.

Para comprobar estas cosas se hizo las pruebas de caja negra que consta de partición de equivalencias, valores limites, transición de estado y tablas de decisión.

Cada de una de ellas son servirán para es correcto funcionamiento ante varios casos que se puede aver dentro del sistema de atención veterinaria.

Gráficos propuestos:

Tabla de Decisión:

Condición	Regla 1	Regla 2	Regla 3	Regla 4
ID válido	Sí	Sí	No	Sí
Nombre del dueño completo	Sí	No	Sí	Sí
Nombre de la mascota completo	Sí	Sí	Sí	No
Especie completa	Sí	Sí	Sí	Sí
Raza completa	Sí	Sí	Sí	Sí
Edad mayor o igual a 0	Sí	Sí	Sí	Sí
Historial médico completo	Sí	Sí	Sí	Sí
Acción	Registrar	No registrar	No registrar	No registrar

Para la tabla de decisión estoy usando las librerías de **pandas** y **matplotlib**, la cual se pone una condición y mediante reglas establecidas del sistema puede mostrarse si se registra o no se registra.

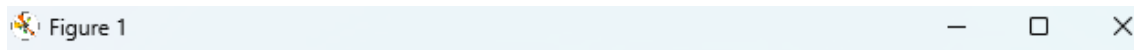
```

1 import unittest
2 from veterinaria import Paciente, SistemaVeterinario
3
4 class TestTablasDecision(unittest.TestCase):
5     def setUp(self):
6         self.sistema = SistemaVeterinario()
7
8     def test_facturacion_decision(self):
9         paciente = Paciente(1, "Juan Perez", "Rex", "Perro", "Lab
10         self.sistema.registrar_paciente(paciente)
11         self.sistema.generar_factura(1, ["Consulta general", "Vac
12         facturas = self.sistema.obtener_facturas()
13         self.assertEqual(facturas[0]["total"], 150.0)
14
15 if __name__ == "__main__":
16     unittest.main()
17

```

Partición de equivalencia:

Utilize la librería matplotlib y pandas para generar el grafico de la tabla.



Parámetro	Clase de Equivalencia	Ejemplo de Valor
ID del Paciente	Válido	123
ID del Paciente	Inválido	-1
ID del Paciente	Inválido	"ABC"
Nombre del Dueño	Válido	"Juan Perez"
Nombre del Dueño	Inválido	""
Nombre de la Mascota	Válido	"Rex"
Nombre de la Mascota	Inválido	""
Especie	Válido	"Perro"
Especie	Inválido	""
Raza	Válido	"Labrador"
Raza	Inválido	""
Edad	Válido	5
Edad	Inválido	-3
Historial Médico	Válido	"Vacunas al día"
Historial Médico	Inválido	""



Para este apartado estoy considerando parámetros, las clases de equivalencia y un ejemplo de valor que se lo puede de considerar de carácter numero, letras, negativos y textos de manera que la información se completa.

```

1 import unittest
2 from veterinaria import Paciente, SistemaVeterinario
3
4 class TestParticionEquivalencia(unittest.TestCase):
5     def setUp(self):
6         self.sistema = SistemaVeterinario()
7
8     def test_registrar_paciente_valido(self):
9         paciente = Paciente(1, "Juan Perez", "Rex", "Perro", "Labrador", 5, "Vacunas al día")
10        resultado = self.sistema.registrar_paciente(paciente)
11        self.assertTrue(resultado)
12        self.assertEqual(self.sistema.obtener_paciente(1), paciente)
13
14    def test_registrar_paciente_invalido(self):
15        paciente = Paciente(None, "", "", "", "", -1, "")
16        resultado = self.sistema.registrar_paciente(paciente)
17        self.assertFalse(resultado)
18        self.assertIsNone(self.sistema.obtener_paciente(None))
19

```

Prueba de transición de estado.

Para este caso se considera los estados y transiciones para nuestro sistema de veterinaria los estado que estoy tomando es.

Estado: Inicio, Registrar Paciente, Gestión de Citas, Gestión de inventario, Gestión de facturación, Salir

Transiciones: Del inicio pasa al estado de registro.

Desde registro del paciente para a la Gestión de citas.

Desde Gestión de Citas para a la Gestión de inventario.

Desde Gestión de inventario a la Gestión de Facturación.

Desde la gestión de facturación a Salir.

```

6 # Definir los estados
7 states = ['Inicio', 'Registrar Paciente', 'Gestión de Citas', 'Ge
8
9 # Añadir nodos para los estados
10 for state in states:
11     dot.node(state, state)
12
13 # Definir las transiciones
14 transitions = [
15     ('Inicio', 'Registrar Paciente'),
16     ('Registrar Paciente', 'Gestión de Citas'),
17     ('Gestión de Citas', 'Gestión de Inventario'),
18     ('Gestión de Inventario', 'Gestión de Facturación'),
19     ('Gestión de Facturación', 'Salir'),
20 ]
21

```

Quiero hacer énfasis en este apartado ya que, se puede poner estado también para Editar, Eliminar empezando desde cuando se registra el paciente hasta la gestión de la factura por que no falta el error humano que puede generar dentro del sistema.

Transición de estado:

Para este grafico utilice la herramienta diagrams.net para crear el diagrama y se pueda interpretar de manera grafica. Ya que tuve inconveniente para la la generación del código en visual.



Valores limites:

Tome en cuenta el peso de la mascota, la edad, la cantidad de citas, cada uno de ellos tienen como valor mínimo y valor máximo, para la aplicación de propósito es buscar errores al pasar esos límites convirtiéndolos en valores críticos, con ello ya viene la implementación y los valores de resultados esperados.