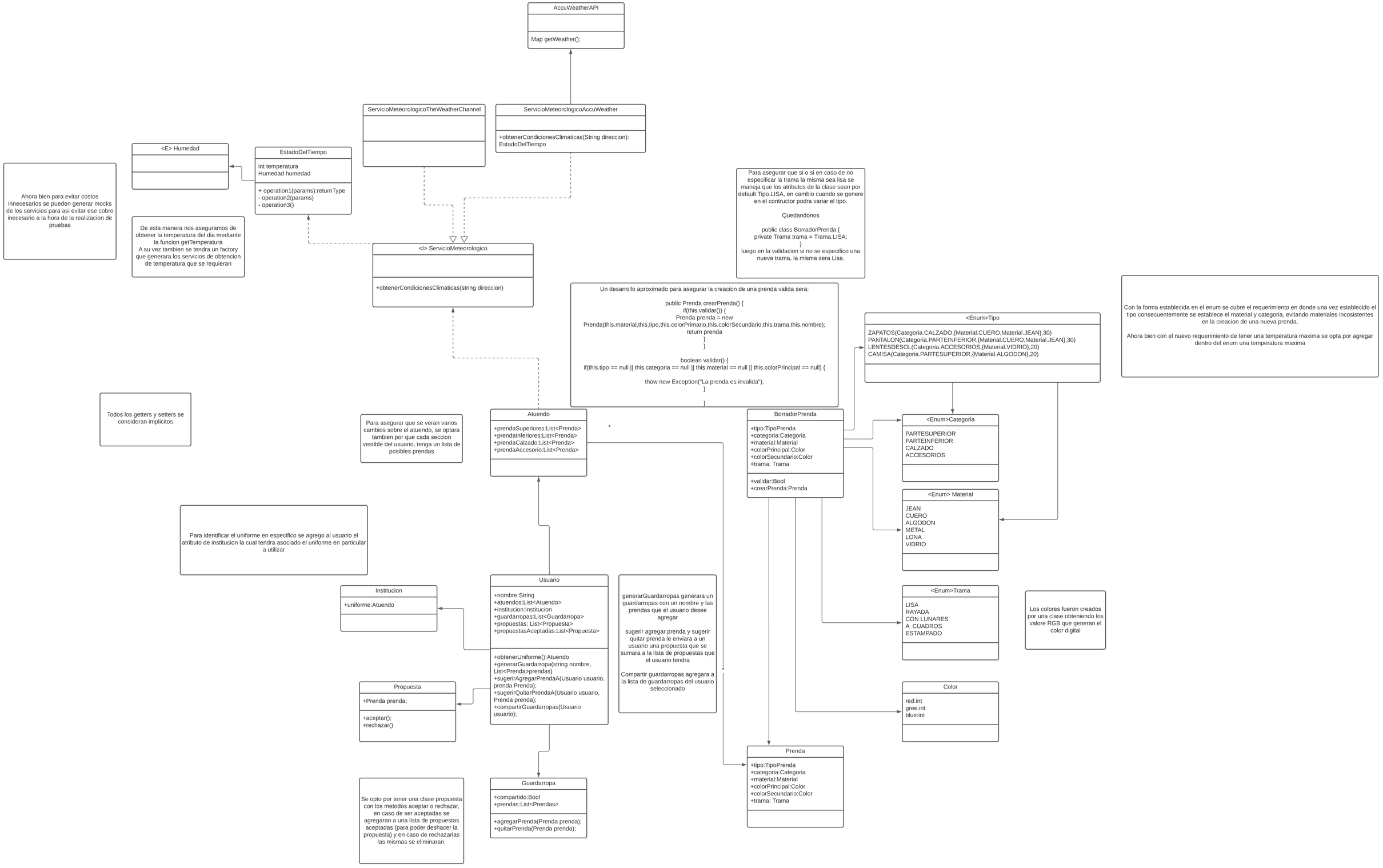
AccuWeatherAPI
Map getWeather();
ServicioMeteorologicoTheWeatherChannel
ServicioMeteorologicoAccuWeather
+obtenerCondicionesClimaticas(String direccion): EstadoDelTiempo
<E> Humedad
EstadoDelTiempo
int temperatura
Humedad humedad
+ operation1(params):returnType
- operation2(params)
- operation3()
Ahora bien para evitar costos innecesarios se pueden generar mocks de los servicios para asi evitar ese cobro inecesario a la hora de la realizacion de pruebas
De esta manera nos aseguramos de obtener la temperatura del dia mediante la funcion getTemperatura
A su vez tambien se tendra un factory que generara los servicios de obtencion de temperatura que se requieran
<I> ServicioMeteorologico
+obtenerCondicionesClimaticas(string direccion)
Para asegurar que si o si en caso de no especificar la trama la misma sea lisa se maneja que los atributos de la clase sean por default Tipo.LISA, en cambio cuando se genere en el contructor podra variar el tipo.
Quedandonos
public class BorradorPrenda {
private Trama trama = Trama.LISA;
}
luego en la validacion si no se especifico una nueva trama, la misma sera Lisa.
Un desarrollo aproximado para asegurar la creacion de una prenda valida sera:
public Prenda crearPrenda() {
if(this.validar()) {
Prenda prenda = new
Prenda(this.material,this,tipo,this.colorPrimario,this.colorSecundario,this.trama,this.nombre);
return prenda
}
}
boolean validar() {
if(this.tipo == null || this.categoria == null || this.material == null || this.colorPrincipal == null) {
thow new Exception("La prenda es invalida");
}
}
<Enum>Tipo
ZAPATOS(Categoria.CALZADO,{Material.CUERO,Material.JEAN},30)
PANTALON(Categoria.PARTEINFERIOR,{Material.CUERO,Material.JEAN},30)
LENTESDESOL(Categoria.ACCESORIOS,{Material.VIDRIO},20)
CAMISA(Categoria.PARTESUPERIOR,{Material.ALGODON},20)
Con la forma establecida en el enum se cubre el requerimiento en donde una vez establecido el tipo consecuentemente se establece el material y categoria, evitando materiales incosistentes en la creacion de una nueva prenda.
Ahora bien con el nuevo requerimiento de tener una temperatura maxima se opta por agregar dentro del enum una temperatura maxima
Todos los getters y setters se consideran implicitos
Para asegurar que se veran varios cambios sobre el atuendo, se optara tambien por que cada seccion vestible del usuario, tenga un lista de posibles prendas
Atuendo
+prendaSuperiores:List<Prenda>
+prendaInferiores:List<Prenda>
+prendaCalzado:List<Prenda>
+prendaAccesorio:List<Prenda>
*
BorradorPrenda
+tipo:TipoPrenda
+categoria:Categoria
+material:Material
+colorPrincipal:Color
+colorSecundario:Color
+trama: Trama
+validar:Bool
+crearPrenda:Prenda
<Enum>Categoria
PARTESUPERIOR
PARTEINFERIOR
CALZADO
ACCESORIOS
<Enum> Material
JEAN
CUERO
ALGODON
METAL
LONA
VIDRIO
Para identificar el uniforme en especifico se agrego al usuario el atributo de institucion la cual tendra asociado el uniforme en particular a utilizar
Usuario
+nombre:String
+atuendos:List<Atuendo>
+institucion:Institucion
+guardarropas:List<Guardarropa>
+propuestas: List<Propuesta>
+propuestasAceptadas:List<Propuesta>
+obtenerUniforme():Atuendo
+generarGuardarropa(string nombre, List<Prenda>prendas)
+sugerirAgregarPrendaA(Usuario usuario, prenda Prenda);
+sugerirQuitarPrendaA(Usuario usuario, Prenda prenda);
+compartirGuardarropas(Usuario usuario);
generarGuardarropas generara un guardarropas con un nombre y las prendas que el usuario desee agregar
sugerir agregar prenda y sugerir quitar prenda le enviara a un usuario una propuesta que se sumara a la lista de propuestas que el usuario tendra
Compartir guardarropas agregara a la lista de guardarropas del usuario seleccionado
Institucion
+uniforme:Atuendo
<Enum>Trama
LISA
RAYADA
CON LUNARES
A CUADROS
ESTAMPADO
Los colores fueron creados por una clase obteniendo los valore RGB que generan el color digital
Propuesta
+Prenda prenda;
+aceptar();
+rechazar()
Color
red:int
gree:int
blue:int
Prenda
+tipo:TipoPrenda
+categoria:Categoria
+material:Material
+colorPrincipal:Color
+colorSecundario:Color
+trama: Trama
Se opto por tener una clase propuesta con los metodos aceptar o rechazar, en caso de ser aceptadas se agregaran a una lista de propuestas aceptadas (para poder deshacer la propuesta) y en caso de rechazarlas las mismas se eliminaran.
Guardarropa
+compartido:Bool
+prendas:List<Prendas>
+agregarPrenda(Prenda prenda);
+quitarPrenda(Prenda prenda);