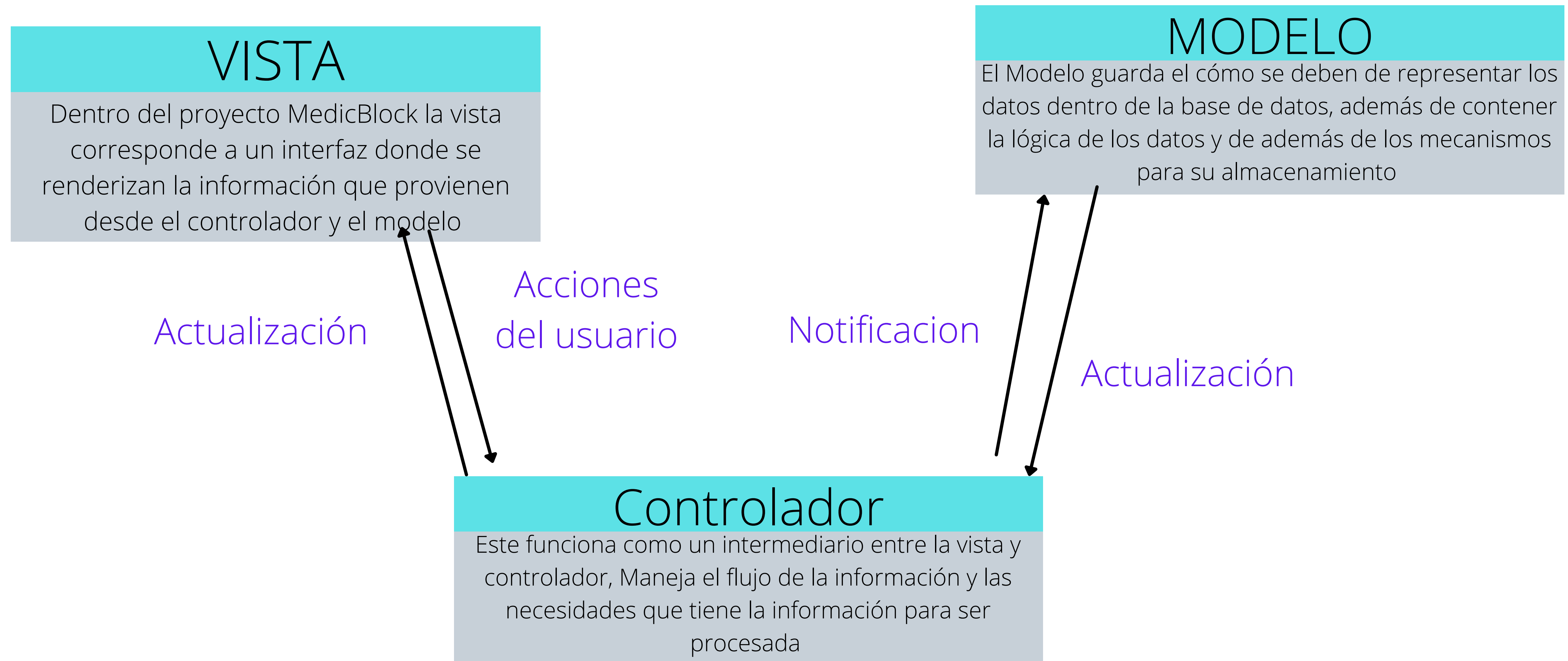


# MVC MedicBlock



# VISTA

Esta vista renderiza/muestra los datos relacionados con la creación de una cita en el proyecto

MedicBlock

HOME

Buscar Paciente

606cbd3792a1fa0038e41cc6

Proposito de la cita \*

Cita poderosa

Nombre: Alejandro Jimenez

Correo: alejandrojimenez1577@gmail.com

Numero de telefono: 4645454565

Fecha

03/04/2022

Hora

09:09 PM

Virutal

Esto generar una reunion virtual en la plataforma zoom

CREAR CITA

# MODELO

El modelo de citas contiene todos los parámetros relacionados a las citas junto con los datos necesarios para su correcto funcionamiento, este también contiene todos los métodos de acceso a la información

```
var Cita = new mongoose.Schema({
  Medico: {type:String,required:true},
  Proposito:{type:String,required:true},
  Paciente:{type:String,required:true},
  FeHo:{type:Date,required:true},
  Virtual:{type:Boolean,required:true},
  ZoomID:{type:Number,required:false},
  ZoomPassword:{type:String,required:false},
  Estado:{type:Number,required:true,default:1},
  HASH:{type:String,requiere:false}
})
```

Modelo de Citas

# CONTROLADOR

El controlador en este caso maneja el flujo de información entre la vista y el modelo, llamando a los métodos del modelo para la información y enviándolo a la vista

```
router.put('/cambiarestadocita', function (req, res) {
  try{
    let CitaID = req.body.CitaID;
    let estado = req.body.Estado;
    if(estado === 0 || estado === 1 || estado === 2)
    {
      cita.findOneAndUpdate({_id: CitaID}, {Estado: estado}, (err,doc)=>{
        if(err){
          return res.status(200).send({Cambio:false,Mensaje:"Estado no actualizado"})
        }
      })
    }
  }
})
```

Controlador de Citas

# VISTA

Esta vista renderiza/muestra los datos de los usuarios registrados y permite alterar los roles del usuario

MedicBlock HOME						
Nombre	Rango	Correo	Telefono	Telefono verificado	Correo verificado	Control
David Garcia Cano	Master	davidfgdh@gmail.com	4641516090	No	No	E8287B0038E75101
David Garcia Cano	Usuario	garciadavid7894@gmail.com	4641516095	No	Si	DBFEF10038E63C42
David Garcia	Paciente	david871ab@gmail.com	4641516098	No	Si	AD3E500038602DE2
Brandon Antonio garcia	Medico	correoperrondelpacha@gmail.com	4641516040	No	No	CDR606A4D03F7B12A00383A11D2
Alejandro Jimenez	Paciente	alejandrojimenez1577@gmail.com	4645454565	No	Si	CDR606CBD3792A1FA0038E41CC6

- Usuario
- Paciente
- Medico
- Master

# MODELO

El modelo de usuario contiene todos los datos importantes de un usuario, desde el registro, maneja a su vez la información de autorización y autenticación dando acceso al controlador a información que solicita.

```
let UserSchema = new mongoose.Schema({
  Nombre: {type:String,required:true},
  Apellido:{type:String,required:true},
  Numerodetelefono:{type:Number,required:true},
  Validoregistro:{type:Boolean, default:false},
  Correo: {type:String,required:true},
  Valicorreo:{type:Boolean, default:false},
  Contraseña: {type:String,required:true},
  Fecharegisto: {type: Date, default: Date.now},
  Medicos:[String],
  Pacientes:[String],
  Rol:{type:Number,default:0},
```

Modelo de usuarios

# CONTROLADOR

El controlador en este caso maneja el flujo de información entre la vista y el modelo, llamando a los métodos del modelo para la información y enviándolo a la vista

```
router.put('/usuariodatos/:modificacion',validar,(req,res)=>{
  try{
    let nombre = req.body.Nombre
    let apellido = req.body.Apellido
    let correo = req.body.Correo
    let tel = req.body.Tel

    if(req.params.modificacion==="nombre"){
      if(nombre!=null) {
        UserDB.findOneAndUpdate({_id: req.id}, {Nombre: nombre}, (err, doc) => {
          if (err) {
```

Controlador de Usuarios

# VISTA

Esta vista permite añadir la información faltante para lograr validar la identidad médica a acceder al rol de médico, esto enviando

MedicBlock

HOME

Datos BasicosDatos MedicosDocumentos de Validacion

Informacion para validacion de identidad medica

Institución

Cédula Profesional

Institución de estudios

Cédula Profesional igual a como parece en el documento

Especialidad

Tel. Consultorio

En caso de tenerla

Numero a 10 digitos

Dirección de Consultorio

ENVIAR

# MODELO

Este modelo almacena la información médica bajo un nivel de cifrado y proporciona acceso a la información

```
Estado:{type:String},
Ciudad:{type:String},
CURP:{type:String},
FechaNacimiento:{type:Date},
INEcargada:{type:Boolean},
Institucion: {type:String},
CedulaProfesional:{type:String},
Especialidad:{type:String},
DireccionConsultorio:{type:String},
TelConsultorio:{type:Number},
AzureStoreDocumentos:{
  INEId:{type:String},
  CompDomiciliosId:{type:String},
  CedulaProfesionalId:{type:String},
  TituloId:{type:String}
},
Documentos:{type:Boolean,default:false}
```

Modelo de Datos Medicos

# CONTROLADOR

El controlador gestiona el flujo de la información entre la vista y el modelo, este responde a los eventos que realiza el usuario (Acciones), generando peticiones al modelo y almacenando

```
router.post('/identidad/medica',
  Validar,
  body('Institucion').isLength({min:5}),
  body('CedulaProfesional').isLength({min:10}),
  body('DireccionConsultorio').isLength({min:5}),
  body('TelConsultorio').isLength({min:10,max:10}).isNumeric(),
  authorize([Role.Admin,Role.Usuario]),
  (req,res)=>{
    const errors = validationResult(req);
    if (!errors.isEmpty()) {
      return res.status(400).json({ errors: errors.array() });
    }
    UserDB.findByIdAndUpdate(req.id,{
      Institucion:req.body.Institucion,
      CedulaProfesional:req.body.CedulaProfesional,
      Especialidad:req.body.Especialidad,
      DireccionConsultorio:req.body.DireccionConsultorio,
      TelConsultorio:req.body.TelConsultorio,
```

Controlador de Datos Medicos