



Comisión de Agua Potable, Alcantarillado y Saneamiento de Uruapan

---

## Manual Técnico.

---

*Desarrollador:*

*Peña Paniagua Sergio Eli.*

*Versión. 1.0*

*Fecha: 21 de agosto del 2024.*

### Índice.

<b>Introducción.....</b>	<b>5</b>
<b>Objetivo.....</b>	<b>6</b>
Objetivo general.....	6
Objetivos específicos.....	6
Público objetivo.....	7
<b>Requerimientos del sistema.....</b>	<b>7</b>
Requerimientos de servidor.....	7
Requerimientos de hardware.....	7
Requerimientos de software.....	7
<b>Diseño Arquitectónico.....</b>	<b>8</b>
Diagrama de clases UML.....	8
Patrón Modelo Vista Controlador. (MVC).....	9
Diagramas de caso de uso.....	10
<b>Diseño de datos.....</b>	<b>14</b>
Diagrama entidad relación.....	14
Diccionario de datos.....	17
<b>Diseño de interfaz.....</b>	<b>35</b>
● Vista: Login y cambiar contraseña.....	35
● Vista: Portal inicial (médico, empleado, gestor-rh).....	36
● Vista: Empleados.....	37
● Vista: Ver empleado.....	38
● Vista: Historial general de solicitudes.....	45
● Vista: Ver solicitud.....	45
● Vista: Configuración.....	45
● Vista: Permisos (administración de usuarios, roles, permisos, asignaciones).....	48
● Vista: Incidencias.....	50
● Vista previa de impresión:.....	59
<b>Desarrollo y personalización.....</b>	<b>62</b>
Vistas.....	62
Site.....	62
- login.....	62
- portalemployeed.....	64
- portalgestionrh.....	68
- portalmedico.....	71
- error.....	73
- cambiarcontraseña.....	74
- configuración.....	75
Empleado.....	77
- _form.....	77

- create.....	82
- index.....	83
- view.....	100
- informacion_personal.....	119
- informacion_laboral.....	131
- informacion_vacacional.....	141
- informacion_solicitudes.....	151
- informacion_consultas.....	160
- documentos_empleado.....	163
- documento_medico_empleado.....	170
- antecedente_hereditario.....	177
- antecedente_patologico.....	181
- antecedente_no_patologico.....	183
- antecedente_ginecologico, antecedente_obstrectico, antecedente_perinatal, alergias, exploracion_fisica, interrogatorio_medico.....	212
- antecedente_no_patologico.....	213
Solicitud.....	213
- index.....	213
- view.....	217
Formatos de incidencias.....	232
- _form.....	232
- view.....	237
- historial / index.....	242
- excel-html.....	246
Modelos.....	251
- Empleado.....	251
- Documento.....	256
- CambioHorarioTrabajo.....	258
- Usuario.....	262
- LoginForm.....	266
Controladores.....	269
- Creación del empleado.....	269
- Actualizar y ver datos del empleado.....	278
- Crear documentos, subir, descargar y visualizar archivos.....	291
- Ver solicitudes.....	293
- Controlador del sitio.....	296
- Controladores de los formatos.....	303
● historial e index.....	303
● view y delete.....	305
● create.....	306
● exportHtml y exportHtmlSegundoCaso.....	309
- Avisos.....	317
Extensiones.....	321

COMPOSER.....	321
RBAC.....	322
AdminLTE3.....	323
Estilos.....	323
<b>Conclusión.....</b>	<b>324</b>

### Introducción.

Sistema web para la gestión de recursos humanos. Este sistema ha sido creado para optimizar diversos procesos en el área de recursos humanos y entre el personal involucrado en la CAPASU. La herramienta permite el registro de información de los empleados, así como la gestión de sus documentos en un expediente digital. Además, incluye un módulo para registrar y gestionar la información médica, conformando un expediente médico digital accesible. El sistema facilita llevar un control de solicitudes para la gestión de permisos, reportes de tiempo extra, citas médicas, entre otros trámites internos.

El portal también ofrece un apartado para que los empleados, utilizando sus credenciales, puedan acceder al sistema y realizar solicitudes de manera sencilla. A través de un formulario fácil de utilizar, el sistema genera automáticamente los archivos con su respectiva plantilla listos para ser impresos.

Adicionalmente, se incluye un módulo que permite al personal médico gestionar las citas médicas solicitadas por los empleados, registrar la información médica correspondiente, y llevar un historial de las consultas.

Este manual proporciona una guía completa para entender la estructura y desarrollo del sistema. En la primera parte, se describen los tres tipos de diseño base, junto con diagramas y otras herramientas visuales, que permiten comprender la arquitectura y los escenarios que maneja este sistema.

Más adelante, se detallan las funciones principales del sistema, explicando cómo está construido el código bajo la arquitectura MVC (Modelo, Vista, Controlador). Esto facilita la identificación de las partes del código relevantes para posibles modificaciones.

Finalmente, se presentan los requerimientos necesarios para el correcto funcionamiento del sistema, incluyendo las extensiones importantes y la descripción de los permisos que se deben tener en cuenta.

### Objetivo.

#### Objetivo general.

El objetivo de este sistema web integral es optimizar la gestión de formatos de incidencias y datos de empleados en la CAPASU, así como proporcionar una mejora al manejo del expediente médico de los trabajadores. Con el propósito de mejorar la eficiencia operativa y la accesibilidad a la información relevante, centralizando los datos y automatizando los procesos actuales.

#### Objetivos específicos.

- Expediente digital de los empleados. Almacenar la información relevante de los empleados de la empresa (información personal, laboral, de contacto, documentos personales y empresariales, historial de solicitudes sobre incidencias).
- Expediente médico de los empleados. Registrar la información médica de los empleados (Antecedentes, exploración física, interrogatorio médico, alergias, documentos médicos y el historial de citas médicas y consultas de estos).
- Citas médicas. Generar solicitudes de citas médicas de los empleados con el médico de la empresa a través del sistema. Y notificar al médico de la empresa de las citas médicas solicitadas por los empleados.
- Consultas médicas. Guardar la información recabada en las consultas médicas de los empleados (la información se registra en el expediente médico del empleado).
- Historial de las solicitudes de incidencias. Recopila la información que se genera con los registros de las solicitudes de incidencias que realizan los empleados.
- Crear solicitudes de incidencias. Generar solicitudes de los diferentes tipos de incidencias que se manejan en la empresa.
- Imprimir solicitudes de incidencias con su respectivo formato. Plasmar la información generada por las solicitudes de incidencias en la plantillas utilizadas por la empresa (llenar las respectivas celdas del excel que se maneja en la plantilla para su posterior conversión a HTML y de ahí poder mandar a imprimir el resultado final).
- Permisos y roles. Asignar los permisos y roles que podrán tener los usuarios (empleados) en el sistema.
- Avisos. Crear avisos que se van a mostrar dentro del sistema a los empleados.
- Manejo de archivos. Subir y almacenar diferentes tipos de archivos.

- Reportes. Generar reportes sobre la información de horas extras de los empleados (Esta información se obtendrá a través del checador; sistema de control de acceso biométrico).

### Público objetivo.

El sistema va dirigido a todo el personal de la Comisión de Agua Potable, Alcantarillado y Saneamiento de Uruapan. (Empleados en general, médico de la empresa y área de recursos humanos).

### Requerimientos del sistema.

#### Requerimientos de servidor.

- PHP 8.3.3 o superior.
- Extensiones de PHP:
  - pdo\_mysql
  - intl
  - mbstring
  - openssl
  - fileinfo
  - cURL
  - zip
  - php\_gd
  - php\_exif
- MYSQL

#### Requerimientos de hardware.

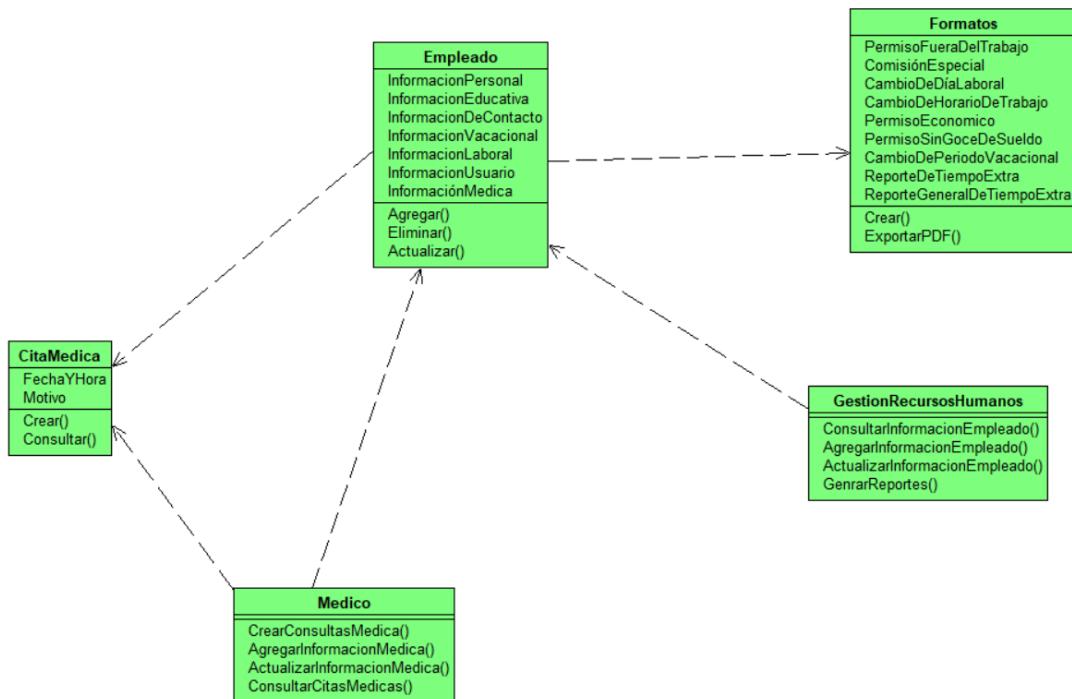
- Ram: 4GB o superior.
- Espacio disponible: 2 GB o superior.
- Disco duro SSD (de preferencia).
- Procesador: Intel i3.

#### Requerimientos de software.

- Sistema operativo Windows Server.
- Composer.

### Diseño Arquitectónico.

#### Diagrama de clases UML.



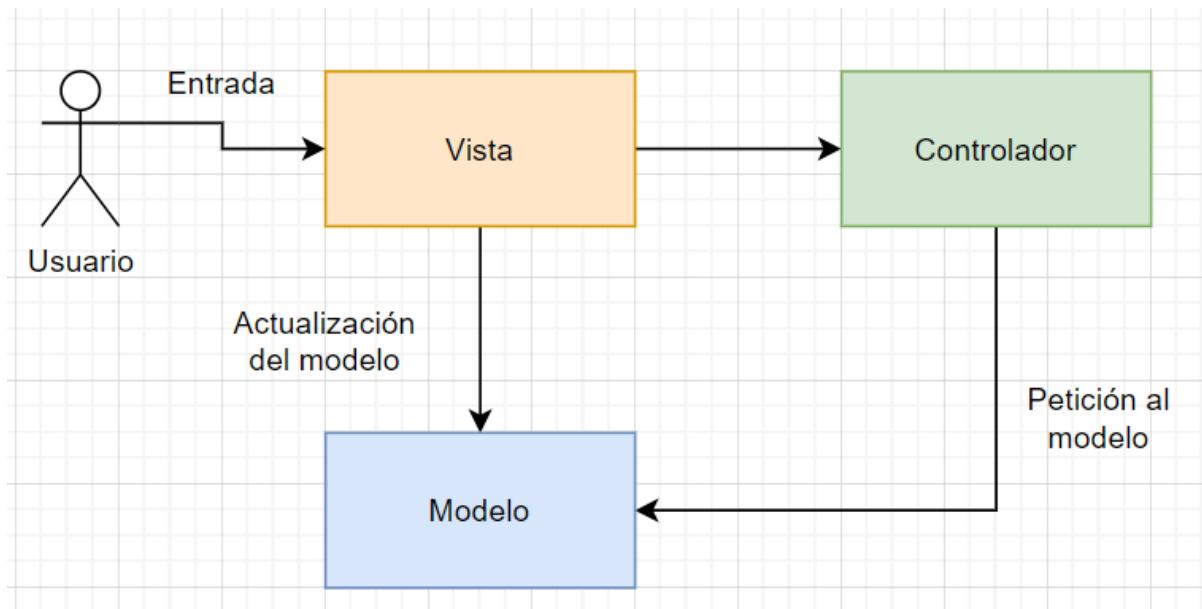
Descripción:

- Clase empleado
  - El empleado cuenta con información personal, educativa, de contacto, vacacional, laboral, de usuario y médica. Este puede ser creado, eliminado y su información actualizada.
- Clase formatos
  - El empleado puede tener acceso a los diferentes tipos de formatos de incidencias para su creación y posteriormente exportarlos a PDF.
- Cita médica
  - El empleado puede crear una cita médica especificando la fecha y

hora de esta, además de incluir el motivo. También puede consultarla.

- Médico
  - El médico de la empresa puede crear consultas médicas, agregar y actualizar la información médica de los empleados. Puede consultar la información de las citas de los empleados, así como el historial de estas.
- Gestión de recursos humanos
  - El empleado encargado de recursos humanos podrá consultar la información completa de los empleados de la empresa y añadir y actualizar dicha información. También podrá generar reportes y consultar el historial de solicitudes de formatos de incidencias que estos generen.

### Patrón Modelo Vista Controlador. (MVC).



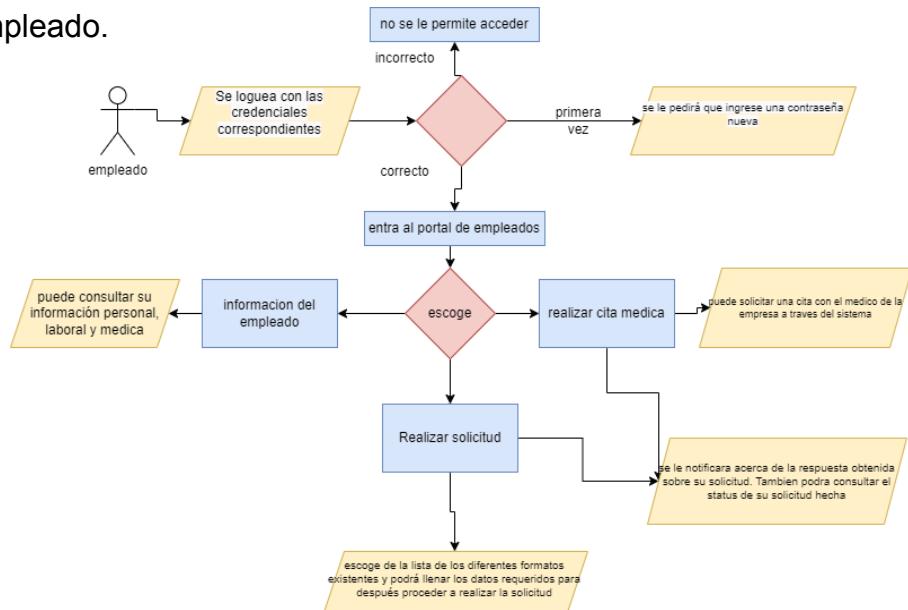
#### Descripción:

- El sistema web se va a estructurar con la arquitectura modelo vista controlador. De forma que el sistema se va a dividir en tres componentes principales, facilitando el desarrollo y mantenimiento de este.

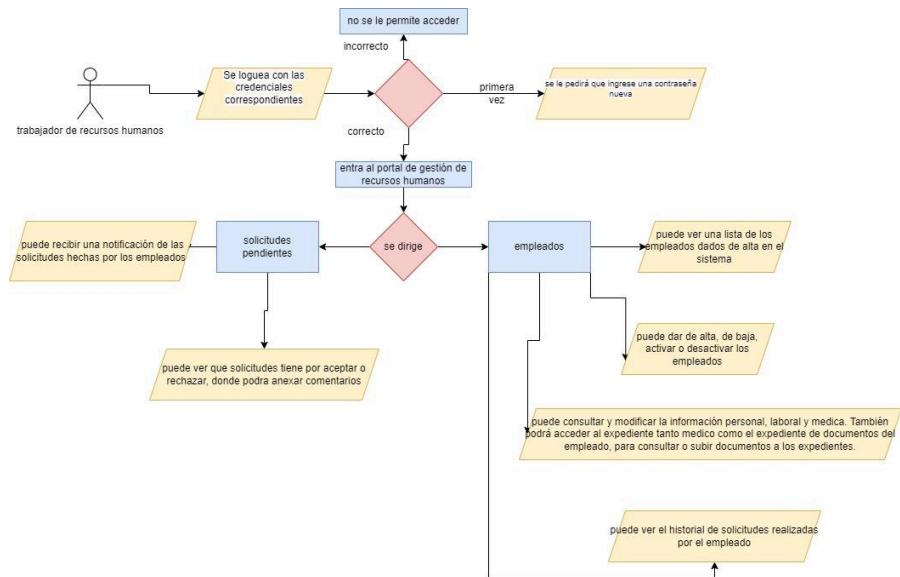
- Modelo. Esta capa gestiona los datos y la lógica de negocio. Aquí se incluyen las operaciones de creación, consultas, actualización y eliminación sobre la base de datos. También se encarga de notificar a las vistas y controladores cuando los datos han cambiado.
- Vista. En esta capa, las vistas son las responsables de generar la interfaz de usuario y mostrar la información proporcionada por el modelo. Es aquí donde los usuarios (empleados y el médico de la empresa) interactúan con el sistema a través de estas vistas donde se incluyen formularios, tablas y más elementos de interfaz de usuario.
- Controlador. Actúa como intermediario entre el modelo y la vista, este recibe las entradas del usuario a través de las vistas (por ejemplo las solicitudes de los formularios), procesa la información y devuelve los resultados a la vista. Se encarga de manejar la lógica del flujo de la aplicación.
- Flujo de trabajo:
  - Interacción con la vista. Los usuarios interactúan con la interfaz del sistema ingresando datos o solicitando información requerida.
  - Acción del controlador. Este recibe la solicitud del usuario, procesa la entrada y realiza las acciones necesarias como consultar o actualizar el modelo.
  - Actualización del modelo. Este realiza las operaciones sobre la base de datos y cualquier otra lógica de negocio necesaria.
  - Actualización de la vista. El controlador actualiza la vista con los datos del modelo, para mostrar al usuario la información solicitada o confirmando que se realizó la acción.

## Diagramas de caso de uso.

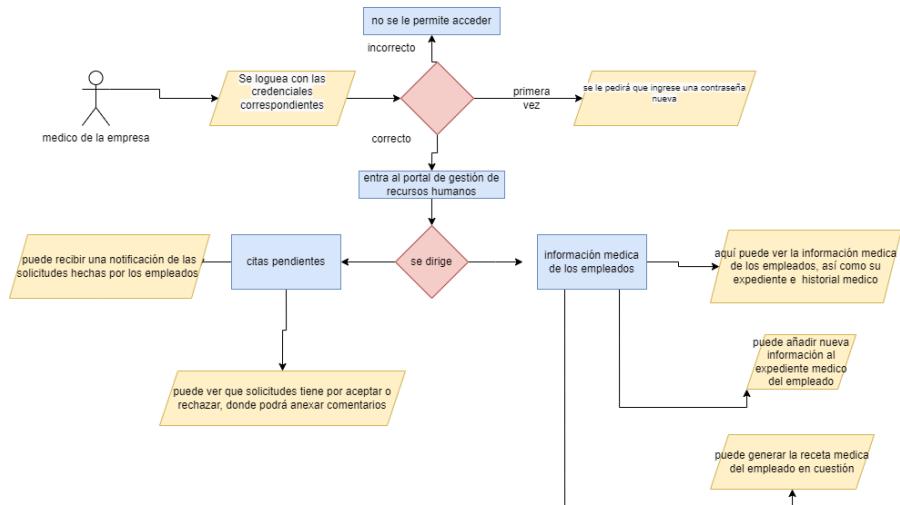
- Empleado.



- Trabajador de recursos humanos.



- Médico de la empresa.



### Descripción.

- En este caso contamos con 3 casos de usuario. El primero en el diagrama es empleado, el cual debe de pasar por un login para poder entrar en el sistema, dicho empleado cuenta con sus credenciales de acceso. Una vez fue validado su inicio de sesión y fue exitoso, si es la primera vez que el usuario inicia sesión, a este se le pide que cambie la contraseña actual y asigne una nueva. Una vez realizado esto, el empleado podrá pasar al portal para empleados. (si no es la primera vez que el usuario inicia sesión, el sistema ya no le pedirá actualizar la contraseña). En el portal empleado puede consultar

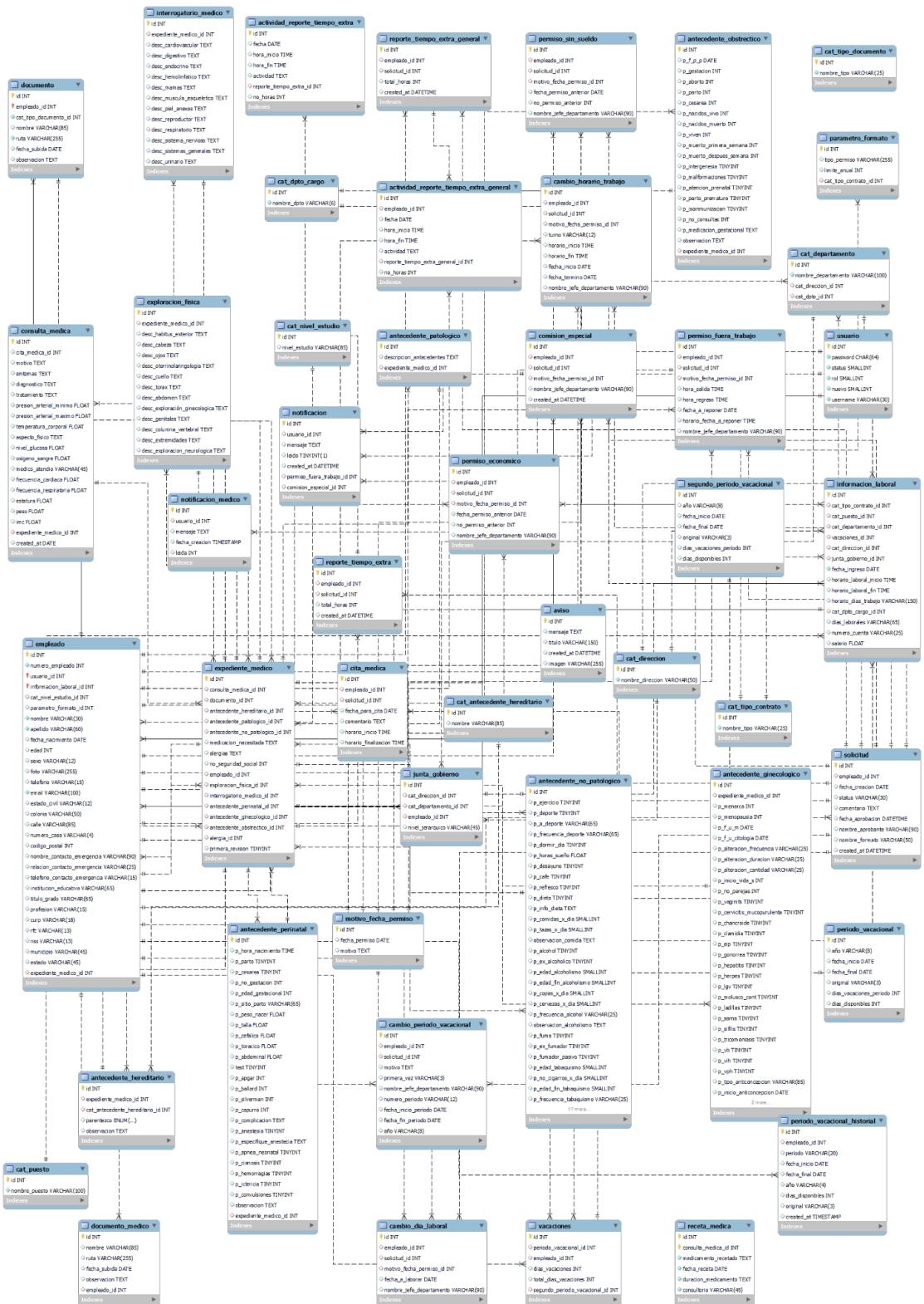
su información personal, laboral y médica, también puede escoger entre los diferentes formatos que maneja la empresa para realizar una solicitud que se va a enviar al trabajador de recursos humanos para aceptar o rechazar la solicitud generada por el empleado. El empleado podrá consultar el estatus que tenga sus solicitudes hechas, también su historial y se le notifica si fue aceptada o rechazada.

En el caso del trabajador de recursos humanos, en cuanto al inicio de sesión tiene el mismo comportamiento inicial que el empleado. Una vez se dio el inicio de sesión, entra al portal de gestión de recursos humanos, aquí puede consultar los empleados dados de alta en el sistema. Puede dar de alta, baja, activar o desactivar empleados, también tiene permitido consultar la información personal, laboral y educativa de estos, así como editarla. Puede ver el expediente y subir los documentos que forman parte del expediente de cada empleado. Puede ver el historial de solicitudes hechas por ellos y también tiene acceso a la información del expediente médico de los empleados, donde podrá subir documentos médicos del empleado. Otra de las acciones más importantes que puede realizar, es aceptar o rechazar las solicitudes generadas por los empleados, donde tendrá la capacidad de si acepta la solicitud poder imprimir el formato correspondiente con los datos del empleado en cuestión y así notificar al empleado que su solicitud fue aceptada, o si fue rechazada notificar y explicar el motivo del rechazo.

En el caso del médico de la empresa, en cuanto al inicio de sesión tiene el mismo comportamiento inicial que el empleado. Una vez se dio el inicio de sesión, puede consultar información acerca de los empleados, también consultar el expediente e historial médico de estos. Otra acción importante que puede realizar es aceptar o rechazar las solicitudes de citas médicas hechas por los empleados, a este se le notificará de las citas médicas que soliciten. También cuenta con la opción de llevar el seguimiento sobre la información médica de los empleados, así como imprimir las recetas y añadir al historial médico del empleado la información obtenida en la cita.

## Diseño de datos.

## Diagrama entidad relación.



### Descripción.

- En este diagrama se observa el diseño y la relación entre tablas que tiene la base de datos.
  - *Tabla usuario*. Almacena los datos del usuario del empleado.
  - *Tabla empleado*. Almacena la información personal y de contacto del empleado.
  - *Tabla cat\_nivel\_estudio*. Almacena los diferentes niveles de estudios.
  - *Tabla informacion\_laboral*. Almacena la información laboral del empleado.
  - *Tabla cat\_departamento*. Almacena los departamentos que existen dentro de la empresa, los cuales forman parte de la información laboral del empleado.
  - *Tabla cat\_dirección*. Almacena las direcciones que hay en la empresa, forman parte de la información laboral del empleado.
  - *Tabla cat\_tipo\_contrato*. Almacena los tipos de contrato que existen en la empresa, forma parte de la información laboral del empleado.
  - *Tabla cat\_puesto*. Almacena los diferentes puestos que hay en la empresa, forma parte de la información laboral del empleado.
  - *Tabla junta\_gobierno*. Almacena qué empleados son jefes o directores de las unidades o direcciones de la empresa.
  - *Tabla expediente*. En esta tabla se almacena información respecto a los documentos de cada empleado.
  - *Tabla cat\_tipo\_documento*. Almacena los diferentes tipos de documentos que manejan en el expediente del empleado.
  - *Tabla permiso\_fuera\_trabajo*. Almacena los datos que necesita este formato.
  - *Tabla comision\_especial*. Almacena los datos que necesita este formato.
  - *Tabla permiso\_economico*. Almacena los datos que necesita este formato.
  - *Tabla permiso\_sin\_sueldo*. Almacena los datos que necesita este formato.
  - *Tabla cambio\_dia\_laboral*. Almacena los datos que necesita este formato.
  - *Tabla cambio\_horario\_trabajo*. Almacena los datos que necesita este formato.
  - *Tabla reporte\_tiempo\_extra*. Almacena los datos que necesita este formato.
  - *Tabla cambio\_periodo\_vacacional*. Almacena los datos que necesita este formato.

- *Tabla motivo\_fecha\_permiso.* Almacena la fecha de permiso y el motivo de los formatos de las incidencias.
- *Tabla expediente\_medico.* Almacena la información médica del empleado.
- *Tabla documento\_medico.* Almacena la información respecto a los documentos médicos del empleado.
- *Tabla cita\_medica.* Almacena información respecto a la solicitud de cita médica que haya generado el empleado.
- *Tabla documento.* Almacena la información de los documentos / archivos personales y laborales del empleado.
- *Tabla interrogatorio\_medico.* Almacena la información acerca de la descripción de diferentes apartados médicos del empleado.
- *Tabla antecedente\_osbrectico.* Almacena la información acerca de los antecedentes obstrecticos del empleado.
- *Tabla cat\_antecedente\_hereditario.* En esta tabla se guardan registros de los diferentes antecedentes hereditarios que puedan existir.
- *Tabla parametro\_formato.* Almacena los valores que establecen los parametros que tienen los diferentes registros de los formatos de las incidencias.
- *Tabla consulta\_medica.* Almacena la información de las consultas médicas que realice el médico de la empresa al empleado.
- *Tabla exploración\_física.* Almacena la información de las descripciones fisicas medicas del empleado.
- *Tabla antecedente\_patologico.* Almacena la información de los antecedentes patológicos del empleado.
- *Tabla solicitud.* Almacena las solicitudes que realice el empleado, en esta tabla se sabe que tipo de solicitud es realizada.
- *Tabla aviso.* Almacena los avisos que se muestran al empleado.
- *Tabla antecedente\_perinatal.* Almacena la información de los antecedentes perinatales del empleado.
- *Tabla vacaciones.* Almacena la información vacacional del empleado.
- *Tabla periodo\_vacacional.* Almacena la información del primer periodo vacacional del empleado.
- *Tabla segundo\_periodo\_vacacional.* Almacena la información del segundo periodo vacacional del empleado.

## Diccionario de datos.

Tabla	Columna	Tipo de dato	Descripción
cambio_horario_trabajo	id	int PRIMARY KEY	Llave principal de la tabla
cambio_horario_trabajo	empleado_id	int	Atributo para hacer referencia al id de la tabla empleado
cambio_horario_trabajo	solicitud_id	int	Atributo para hacer referencia al id de la tabla solicitud
cambio_horario_trabajo	motivo_fecha_permiso_id	int	Atributo para hacer referencia al id de la tabla motivo_fecha_permiso
cambio_horario_trabajo	turno	varchar(12)	Atributo para registrar turno vespertino o matutino
cambio_horario_trabajo	horario_inicio	time	Atributo para registrar la hora de inicio
cambio_horario_trabajo	horario_fin	time	Atributo para registrar la hora de fin
cambio_horario_trabajo	fecha_inicio	date	Atributo para registrar la fecha inicial
cambio_horario_trabajo	fecha_termino	date	Atributo para registrar la fecha final
cambio_horario_trabajo	nombre_jefe_departamento	varchar(90)	Atributo para registrar el nombre del jefe de departamento
cambio_horario_trabajo	fk_empleado_cambio_horario	FOREIGN KEY	Llave foranea a tabla empleado
cambio_horario_trabajo	fk_motivo_fecha_cambio_horario	FOREIGN KEY	Llave foranea a tabla motivo_fecha_permiso
cambio_horario_trabajo	fk_solicitud_cambio_horario	FOREIGN KEY	Llave foranea a tabla solicitud
cambio_periodo_vacacional	id	int PRIMARY KEY	Llave principal de la tabla
cambio_periodo_vacacional	empleado_id	int	Atributo para hacer referencia al id de la tabla empleado

cambio_periodo_vacacional	solicitud_id	int	Atributo para hacer referencia al id de la tabla solicitud
cambio_periodo_vacacional	motivo	text	Atributo para registrar el texto del motivo
cambio_periodo_vacacional	primera_vez	varchar(3)	Atributo para registrar Si o No
cambio_periodo_vacacional	nombre_jefe_departamento	varchar(90)	Atributo para registrar el nombre del jefe de departamento
cambio_periodo_vacacional	numero_periodo	varchar(12)	Atributo para registrar si es primer o segundo periodo vacacional
cambio_periodo_vacacional	fecha_inicio_periodo	date	Atributo para registrar la fecha inicial
cambio_periodo_vacacional	fecha_fin_periodo	date	Atributo para registrar la fecha final
cambio_periodo_vacacional	año	varchar(8)	Atributo para registrar el año
cambio_periodo_vacacional	fk_empleado_cambio_periodo	FOREIGN KEY	Llave foranea a tabla empleado
cat_actividad_fisica	id	int PRIMARY KEY	Llave principal de la tabla
cat_actividad_fisica	nombre_actividad	varchar(85)	Atributo para registrar el nombre de la actividad fisica
cat_direccion	id	int PRIMARY KEY	Llave principal de la tabla
cat_direccion	nombre_direccion	varchar(50)	Atributo para registrar el nombre de la dirección
cat_dpto_cargo	id	int PRIMARY KEY	Llave principal de la tabla
cat_dpto_cargo	nombre_dpto	varchar(6)	Atributo para registrar las abreviaciones DPTO
cat_nivel_estudio	id	int PRIMARY KEY	Llave principal de la tabla
cat_nivel_estudio	nivel_estudio	varchar(85)	Atributo para registrar los diferentes niveles de estudio

cat_puesto	id	int PRIMARY KEY	Llave principal de la tabla
cat_puesto	nombre_puesto	varchar(100)	Atributo para registrar el nombre del puesto
cat_tipo_contrato	id	int PRIMARY KEY	Llave principal de la tabla
cat_tipo_contrato	nombre_tipo	varchar(25)	Atributo para registrar el tipo de contrato
cat_tipo_documento	id	int PRIMARY KEY	Llave principal de la tabla
cat_tipo_documento	nombre_tipo	varchar(25)	Atributo para registrar el tipo de documento
cita_medica	id	int	Llave principal de la tabla
cita_medica	empleado_id	int	Atributo para hacer referencia al id de la tabla empleado
cita_medica	solicitud_id	int	Atributo para hacer referencia al id de la tabla solicitud
cita_medica	fecha_para_cita	date	Atributo para registrar la fecha de la cita
cita_medica	comentario	text	Atributo para registrar comentarios
cita_medica	horario_cita	time	Atributo para registrar el horario de la cita
cita_medica	fk_solicitud_cita_medica	FOREIGN KEY	Llave foranea a tabla empleado
cita_medica	fk_empleado_cita_medica	FOREIGN KEY	Llave foranea a tabla cita_medica
comision_especial	id	PRIMARY KEY	Llave principal de la tabla
comision_especial	empleado_id	int	Atributo para hacer referencia al id de la tabla empleado
comision_especial	solicitud_id	int	Atributo para hacer referencia al id de la tabla solicitud

comision_especial	motivo_fecha_permiso_id	int	Atributo para hacer referencia al id de la tabla motivo_fecha_permiso
comision_especial	nombre_jefe_departamento	varchar(90)	Atributo para registrar el nombre del jefe de departamento
comision_especial	fk_empleado_comision	FOREIGN KEY	Llave foranea a tabla empleado
comision_especial	fk_motivo_fecha_permiso_comision	FOREIGN KEY	Llave foranea a tabla motivo_fecha_permiso
comision_especial	fk_solicitud_comision	FOREIGN KEY	Llave foranea a tabla solicitud
consulta_medica	id	int PRIMARY KEY	Llave principal de la tabla
consulta_medica	cita_medica_id	int	Atributo para hacer referencia al id de la tabla cita_medica
consulta_medica	motivo	text	Atributo para registrar el texto del motivo
consulta_medica	sintomas	text	Atributo para registrar los sintomas
consulta_medica	diagnostico	text	Atributo para registrar el diagnostico
consulta_medica	tratamiento	text	Atributo para registrar el tratamiento
consulta_medica	presion_arterial_minimo	float	Atributo para registrar el minimo de la presion arterial
consulta_medica	presion_arterial_maximo	float	Atributo para registrar el maximo de la presion arterial
consulta_medica	temperatura_corporal	float	Atributo para registrar la temperatura corporal
consulta_medica	aspecto_fisico	text	Atributo para almacenar aspecto fisico
consulta_medica	nivel_glucosa	float	Atributo para almacenar nivel glucosa
consulta_medica	oxigeno_sangre	float	Atributo para almacenar oxigeno sangre
consulta_medica	medico_atendio	varchar(45)	Atributo para almacenar el medico que atendio

consulta_medica	frecuencia_cardiaca	float	Atributo para almacenar frecuencia cardiaca
consulta_medica	frecuencia_respiratoria	float	Atributo para almacenar frecuencia respiratoria
consulta_medica	estatura	float	Atributo para almacenar estatura
consulta_medica	peso	float	Atributo para almacenar peso
consulta_medica	imc	float	Atributo para almacenar imc
consulta_medica	fk_consulta_cita_medica	FOREING KEY	Clave foránea que referencia a la tabla relacionada con cita_medica
documento	id	int	Llave principal de la tabla
documento	empleado_id	int	Atributo para hacer referencia al id de la tabla empleado
documento	cat_tipo_documento_id	int	Atributo para hacer referencia al id de la tabla cat_tipo_documento
documento	nombre	varchar(85)	Atributo para almacenar nombre
documento	ruta	varchar(255)	Atributo para almacenar ruta
documento	fecha_subida	date	Atributo para almacenar la fecha de subida
documento	observacion	text	Atributo para almacenar observacion
documento	fk_cat_tipo_documento	FOREING KEY	Clave foránea que referencia a la tabla cat_tipo_documento
documento	fk_empleado	FOREIGN KEY	Clave foránea que referencia a la tabla empleado
empleado	id	int	Llave principal de la tabla
empleado	numero_empleado	int	Atributo para almacenar numero empleado
empleado	usuario_id	int	Atributo para hacer referencia al id de la tabla usuario
empleado	informacion_laboral_id	int	Atributo para hacer referencia al id de la tabla informacion_laboral

empleado	cat_nivel_estudio_id	int	Atributo para hacer referencia al id de la tabla cat_nivel_estudio
empleado	parametro_formato_id	int	Atributo para hacer referencia al id de la tabla parametro_formato
empleado	nombre	varchar(30)	Atributo para almacenar el nombre
empleado	apellido	varchar(60)	Atributo para almacenar el apellido
empleado	fecha_nacimiento	date	Atributo para almacenar la fecha nacimiento
empleado	edad	int	Atributo para almacenar la edad
empleado	sexo	varchar(12)	Atributo para almacenar el sexo
empleado	foto	varchar(100)	Atributo para almacenar foto del empleado
empleado	telefono	varchar(15)	Atributo para almacenar el telefono
empleado	email	varchar(100)	Atributo para almacenar el email
empleado	estado_civil	varchar(12)	Atributo para almacenar el estado civil
empleado	colonia	varchar(50)	Atributo para almacenar la colonia
empleado	calle	varchar(85)	Atributo para almacenar la calle
empleado	numero_casa	varchar(4)	Atributo para almacenar el numero casa
empleado	codigo_postal	int	Atributo para almacenar el codigo postal
empleado	nombre_contacto_emergencia	varchar(90)	Atributo para almacenar el nombre del contacto emergencia
empleado	relacion_contacto_emergencia	varchar(25)	Atributo para almacenar relacion contacto emergencia
empleado	telefono_contacto_emergencia	varchar(15)	Atributo para almacenar el telefono del contacto emergencia
empleado	institucion_educativa	varchar(65)	Atributo para almacenar el nombre de la institucion educativa

empleado	profesion	varchar(15)	Atributo para almacenar la profesion/titulo del empleado
empleado	curp	varchar(18)	Atributo para almacenar la curp
empleado	rfc	varchar(13)	Atributo para almacenar el rfc
empleado	nss	varchar(13)	Atributo para almacenar el nss
empleado	municipio	varchar(45)	Atributo para almacenar el municipio
empleado	estado	varchar(45)	Atributo para almacenar el estado
empleado	fk_informacion_laboral	FOREIGN KEY	Clave foránea que referencia a la tabla informacion_laboral
empleado	fk_cat_nivel_estudio	FOREIGN KEY	Clave foránea que referencia a la tabla cat_nivel_estudio
empleado	fk_parametro_formato	FOREIGN KEY	Clave foránea que referencia a la tabla parametro_formato
empleado	fk_usuario	FOREIGN KEY	Clave foránea que referencia a la tabla usuario
expediente_medico	id	int	Llave principal de la tabla
expediente_medico	empleado_id	int	Atributo para hacer referencia al id de la tabla empleado
expediente_medico	consulta_medica_id	int	Atributo para hacer referencia al id de la tabla consultada_medica
expediente_medico	documento_id	int	Atributo para hacer referencia al id de la tabla documento
expediente_medico	antecedente_hereditario_id	int	Atributo para hacer referencia al id de la tabla antecedente_hereditario
expediente_medico	antecedente_patologico_id	int	Atributo para hacer referencia al id de la tabla antecedente_patologico
expediente_medico	medicacion_necesitada	text	Atributo para almacenar la medicacion asignada
expediente_medico	alergias	text	Atributo para almacenar las alergias
expediente_medico	no_seguridad_social	int	Atributo para almacenar el numero de seguro social

expediente_medico	consulta_medica_id	int	Atributo para hacer referencia al id de la tabla expediente_medico
expediente_medico	fk_empleado_exp_medico	FOREING KEY	Clave foránea que referencia a la tabla empleado
expediente_medico	fk_documento_exp_medico	FOREING KEY	Clave foránea que referencia a la tabla documento
expediente_medico	fk_antecedente_patologico	FOREING KEY	Clave foránea que referencia a la tabla antecedente_patologico
expediente_medico	fk_antecedente_hereditario	FOREING KEY	Clave foránea que referencia a la tabla antecedente_hereditario
expediente_medico	fk_consulta_exp_medico	FOREING KEY	Clave foránea que referencia a la tabla consulta_medica
informacion_laboral	id	int PRIMARY KEY	Llave principal de la tabla
informacion_laboral	cat_tipo_contrato_id	int	Atributo para hacer referencia al id de la tabla cat_tipo_contrato
informacion_laboral	cat_puesto_id	int	Atributo para hacer referencia al id de la tabla cat_puesto
informacion_laboral	cat_departamento_id	int	Atributo para hacer referencia al id de la tabla cat_departamento
informacion_laboral	vacaciones_id	int	Atributo para hacer referencia al id de la tabla vacaciones
informacion_laboral	cat_direccion_id	int	Atributo para hacer referencia al id de la tabla cat_direccion
informacion_laboral	jefes_directores_id	int	Atributo para hacer referencia al id de la tabla jefes_directores
informacion_laboral	fecha_ingreso	date	Atributo para almacenar la fecha ingreso
informacion_laboral	horario_laboral_inicio	time	Atributo para almacenar el inicio del horario laboral
informacion_laboral	horario_laboral_fin	time	Atributo para almacenar el fin del horario laboral
informacion_laboral	cat_dpto_cargo_id	int	Atributo para hacer referencia al id de la tabla cat_dpto_cargo
informacion_laboral	dias_laborales	varchar(65)	Atributo para almacenar los dias laborales

informacion_laboral	numero_cuenta	varchar(25)	Atributo para almacenar el numero cuenta
informacion_laboral	salario	float	Atributo para almacenar el salario
informacion_laboral	fk_cat_contrato	FOREIGN KEY	Clave foránea que referencia a la tabla cat_contrato
informacion_laboral	fk_cat_departamento	FOREIGN KEY	Clave foránea que referencia a la tabla cat_departamento
informacion_laboral	fk_cat_direccion	FOREIGN KEY	Clave foránea que referencia a la tabla cat_direccion
informacion_laboral	fk_cat_dpto	FOREIGN KEY	Clave foránea que referencia a la tabla cat_dpto
informacion_laboral	fk_cat_puesto	FOREIGN KEY	Clave foránea que referencia a la tabla cat_puesto
informacion_laboral	fk_jefes_directores	FOREIGN KEY	Clave foránea que referencia a la tabla jefes_directores
informacion_laboral	fk_vacaciones_informacion_laboral	FOREIGN KEY	Clave foránea que referencia a la tabla vacaciones
jefes_directores	id	int PRIMARY KEY	Llave principal de la tabla
jefes_directores	cat_direccion_id	int	Atributo para hacer referencia al id de la tabla cat_direccion
jefes_directores	cat_departamento_id	int	Atributo para hacer referencia al id de la tabla cat_departamento
jefes_directores	empleado_id	int	Atributo para hacer referencia al id de la tabla empleado
jefes_directores	nivel_jerarquico	varchar(45)	Atributo para almacenar el tipo de nivel jerarquico
jefes_directores	fk_departamento	FOREIGN KEY	Clave foránea que referencia a la tabla cat_departamento
jefes_directores	fk_direccion	FOREIGN KEY	Clave foránea que referencia a la tabla cat_direccion
jefes_directores	fk_empleado_junta	FOREIGN KEY	Clave foránea que referencia a la tabla empleado
motivo_fecha_permiso	id	int PRIMARY KEY	Llave principal de la tabla

motivo_fecha_permiso	fecha_permiso	date	Atributo para almacenar la fecha del permiso
motivo_fecha_permiso	motivo	text	Atributo para almacenar el texto del motivo
notificacion	id	int PRIMARY KEY	Llave principal de la tabla
notificacion	usuario_id	int	Atributo para hacer referencia al id de la tabla empleado
notificacion	mensaje	text	Atributo para almacenar el texto del mensaje
notificacion	leido	tinyint(1)	Atributo para hacer referencia al id de la tabla notificacion
notificacion	created_at	datetime	Atributo para almacenar la fecha de creación de la notificación
notificacion	fk_usuario_notificacion	FOREIGN KEY	Clave foránea que referencia a la tabla usuario
parametro_formato	id	int PRIMARY KEY	Llave principal de la tabla
parametro_formato	solicitud_id	int	Atributo para hacer referencia al id de la tabla solicitud
parametro_formato	nombre_formato	varchar(65)	Atributo para almacenar el nombre del formato
parametro_formato	disponible	int	Atributo para almacenar la disponibilidad
parametro_formato	fk_solicitud_parametro	FOREING KEY	Clave foránea que referencia a la tabla relacionada solicitud
periodo_vacacional	id	int PRIMARY KEY	Llave principal de la tabla
periodo_vacacional	año	varchar(8)	Atributo para almacenar el año
periodo_vacacional	fecha_inicio	date	Atributo para almacenar la fecha de inicio
periodo_vacacional	fecha_final	date	Atributo para almacenar la fecha de fin
periodo_vacacional	original	varchar(3)	Atributo para almacenar Si es original o No

periodo_vacacional	dias_vacaciones_periodo	int	Atributo para almacenar los dias de vacaciones del periodo
periodo_vacacional	dias_disponibles	int	Atributo para almacenar los dias disponibles
periodo_vacacional_historial	id	int PRIMARY KEY	Llave principal de la tabla
periodo_vacacional_historial	empleado_id	int	Atributo para hacer referencia al id de la tabla empleado
periodo_vacacional_historial	periodo	varchar(20)	Atributo para almacenar si es primero o segundo periodo
periodo_vacacional_historial	fecha_inicio	date	Atributo para almacenar la fecha de inicio
periodo_vacacional_historial	fecha_final	date	Atributo para almacenar la fecha de fin
periodo_vacacional_historial	año	varchar(4)	Atributo para almacenar el año
periodo_vacacional_historial	dias_disponibles	int	Atributo para almacenar los dias disponibles
periodo_vacacional_historial	original	varchar(3)	Atributo para almacenar Si es original o No
periodo_vacacional_historial	created_at	timestamp	Atributo para almacenar la fecha de creación
periodo_vacacional_historial	fk_empleado_historial_periodos	FOREIGN KEY	Clave foránea que referencia a la tabla relacionada con empleado
permiso_economico	id	int PRIMARY KEY	Llave principal de la tabla
permiso_economico	empleado_id	int	Atributo para hacer referencia al id de la tabla empleado
permiso_economico	solicitud_id	int	Atributo para hacer referencia al id de la tabla solicitud
permiso_economico	motivo_fecha_permiso_id	int	Atributo para hacer referencia al id de la tabla motivo_fecha_permiso
permiso_economico	fecha_permiso_anterior	date	Atributo para almacenar la fecha del permiso anterior
permiso_economico	no_permiso_anterior	int	Atributo para almacenar el no de permiso anterior

permiso_economico	nombre_jefe_departamento	varchar(90)	Atributo para almacenar el nombre del jefe de departamento
permiso_economico	fk_empleado_permiso_economico	FOREIGN KEY	Clave foránea que referencia a la tabla relacionada empleado
permiso_economico	fk_motivo_fecha_permiso_economico	FOREIGN KEY	Clave foránea que referencia a la tabla relacionada motivo_fecha_permiso
permiso_economico	fk_solicitud_permiso_economico	FOREIGN KEY	Clave foránea que referencia a la tabla relacionada con solicitud
permiso_fuera_trabajo	id	int PRIMARY KEY	Llave principal de la tabla
permiso_fuera_trabajo	empleado_id	int	Atributo para hacer referencia al id de la tabla empleado
permiso_fuera_trabajo	solicitud_id	int	Atributo para hacer referencia al id de la tabla solicitud
permiso_fuera_trabajo	motivo_fecha_permiso_id	int	Atributo para hacer referencia al id de la tabla motivo_fecha_permiso
permiso_fuera_trabajo	hora_salida	time	Atributo para almacenar la hora de salida
permiso_fuera_trabajo	hora_Regreso	time	Atributo para almacenar la hora de regreso
permiso_fuera_trabajo	fecha_a_reponer	date	Atributo para almacenar la fecha en la que va a reponer
permiso_fuera_trabajo	horario_fecha_a_reponer	time	Atributo para almacenar horario de la fecha a reponer
permiso_fuera_trabajo	nombre_jefe_departamento	varchar(90)	Atributo para almacenar el nombre del jefe de departamento
permiso_fuera_trabajo	fk_motivo_fecha_permiso	FOREIGN KEY	Clave foránea que referencia a la tabla relacionada motivo_fecha_permiso
permiso_fuera_trabajo	fk_pft_empleado	FOREIGN KEY	Clave foránea que referencia a la tabla relacionada empleado
permiso_fuera_trabajo	fk_solicitud_pft	FOREIGN KEY	Clave foránea que referencia a la tabla relacionada solicitud
permiso_sin_sueldo	id	int PRIMARY KEY	Llave principal de la tabla

permiso_sin_sueldo	empleado_id	int	Atributo para hacer referencia al id de la tabla empleado
permiso_sin_sueldo	solicitud_id	int	Atributo para hacer referencia al id de la tabla solicitud
permiso_sin_sueldo	motivo_fecha_permiso_id	int	Atributo para hacer referencia al id de la tabla motivo_fecha_permiso
permiso_sin_sueldo	fecha_permiso_anterior	date	Atributo para almacenar la fecha del permiso anterior
permiso_sin_sueldo	no_permiso_anterior	int	Atributo para almacenar el numero de permiso anterior
permiso_sin_sueldo	nombre_jefe_departamento	varchar(90)	Atributo para almacenar el nombre del jefe de departamento
permiso_sin_sueldo	fk_empleado_permiso_sin_sueldo	FOREIGN KEY	Clave foránea que referencia a la tabla relacionada empleado
permiso_sin_sueldo	fk_motivo_fecha_permiso_sin_sueldo	FOREIGN KEY	Clave foránea que referencia a la tabla motivo_fecha_permiso
permiso_sin_sueldo	fk_solicitud_permiso_sin_sueldo	FOREIGN KEY	Clave foránea que referencia a la tabla solicitud
receta_medica	id	int PRIMARY KEY	Llave principal de la tabla
receta_medica	consulta_medica_id	int	Atributo para hacer referencia al id de la tabla consultada_medica
receta_medica	medicamento_recetado	text	Atributo para almacenar el medicamento recetado
receta_medica	fecha_receta	date	Atributo para almacenar la fecha de la receta
receta_medica	duracion_medicamento	text	Atributo para almacenar la duracion del medicamento
receta_medica	consultorio	varchar(45)	Atributo para almacenar nombre del consultorio
receta_medica	fk_consulta_receta_medica	FOREING KEY	Clave foránea que referencia a la tabla relacionada con consulta
reporte_tiempo_extra	id	int PRIMARY KEY	Llave principal de la tabla

reporte_tiempo_extra	empleado_id	int	Atributo para hacer referencia al id de la tabla empleado
reporte_tiempo_extra	solicitud_id	int	Atributo para hacer referencia al id de la tabla solicitud
reporte_tiempo_extra	fecha	date	Atributo para almacenar la fecha
reporte_tiempo_extra	horario_inicio	time	Atributo para almacenar el horario de inicio
reporte_tiempo_extra	horario_fin	time	Atributo para almacenar el horario de termino
reporte_tiempo_extra	no_horas	int	Atributo para almacenar el numero de horas
reporte_tiempo_extra	actividad	text	Atributo para almacenar la actividad realizada
reporte_tiempo_extra	total_horas	int	Atributo para almacenar el total de horas
reporte_tiempo_extra	tipo_formato	smallint	Atributo para almacenar el tipo de formato
reporte_tiempo_extra	fk_reporte_tiempo_empleado	FOREING KEY	Clave foránea que referencia a la tabla empleado
reporte_tiempo_extra	fk_reporte_tiempo_solicitud	FOREING KEY	Clave foránea que referencia a la tabla solicitud
segundo_periodo_vacacional	id	int PRIMARY KEY	Llave principal de la tabla
segundo_periodo_vacacional	año	varchar(8)	Atributo para almacenar el año
segundo_periodo_vacacional	fecha_inicio	date	Atributo para almacenar la fecha de inicio
segundo_periodo_vacacional	fecha_final	date	Atributo para almacenar la fecha de fin
segundo_periodo_vacacional	original	varchar(3)	Atributo para almacenar Si es original o No
segundo_periodo_vacacional	dias_vacaciones_periodo	int	Atributo para almacenar los dias de vacaciones del periodo
segundo_periodo_vacacional	dias_disponibles	int	Atributo para almacenar los dias disponibles

solicitud	id	int PRIMARY KEY	Llave principal de la tabla
solicitud	empleado_id	int	Atributo para hacer referencia al id de la tabla empleado
solicitud	fecha_creacion	date	Atributo para almacenar la fecha de creación
solicitud	status	varchar(30)	Atributo para almacenar el status
solicitud	comentario	text	Atributo para almacenar el texto del comentario
solicitud	fecha_aprobacion	datetime	Atributo para almacenar la fecha de aprobacion
solicitud	nombre_aprobante	varchar(90)	Atributo para almacenar el nombre del aprobante
solicitud	nombre_formato	varchar(50)	Atributo para almacenar el nombre del formato
solicitud	fk_solicitud_empleado	FOREING KEY	Clave foránea que referencia a la tabla relacionada empleado
usuario	id	int PRIMARY KEY	Llave principal de la tabla
usuario	password	char(64)	Atributo para almacenar la contraseña
usuario	status	smallint	Atributo para almacenar el status del usuario
usuario	rol	smallint	Atributo para almacenar el rol
usuario	nuevo	smallint	Atributo para almacenar Si es nuevo o No
usuario	username	varchar(30)	Atributo para almacenar el nombre del usuario
vacaciones	id	int PRIMARY KEY	Llave principal de la tabla
vacaciones	periodo_vacacional_id	int	Atributo para hacer referencia al id de la tabla periodo_vacacional
vacaciones	empleado_id	int	Atributo para hacer referencia al id de la tabla empleado

vacaciones	dias_vacaciones	int	Atributo para almacenar los dias de vacaciones
vacaciones	total_dias_vacaciones	int	Atributo para almacenar el total de dias de vacaciones
vacaciones	segundo_periodo_vacacional_id	int	Atributo para hacer referencia al id de la tabla segundo_periodo_vacacional
vacaciones	fk_empleado_vacaciones	FOREIGN KEY	Clave foránea que referencia a la tabla relacionada empleado
vacaciones	fk_periodo_vacacional	FOREIGN KEY	FOREIGN KEY
vacaciones	fk_segundo_periodo_vacacional	FOREIGN KEY	Clave foránea que referencia a la tabla relacionada segundo_periodo_vacacional
aviso	id	int PRIMARY KEY	Llave principal de la tabla.
aviso	mensaje	text	Atributo para almacenar el texto del mensaje del aviso.
aviso	título	varchar(150)	Atributo para almacenar el título del aviso.
aviso	created_at	datetime	Atributo para almacenar la fecha y hora de creación del registro.
aviso	imagen	varchar(255)	Atributo para almacenar la ruta de la imagen del aviso.
alergia	id	int PRIMARY KEY	Llave principal de la tabla.
alergia	p_alergia	text	Atributo para almacenar el texto de la descripción de las alergias.
alergia	expediente_medico_id	int	Atributo para hacer referencia al id de la tabla expediente_medico
alergia	fk_expediente_medico	FOREIGN KEY	Clave foránea que referencia a la tabla relacionada expediente_medico
antecedente_ginecologico	id	int PRIMARY KEY	Llave principal de la tabla.

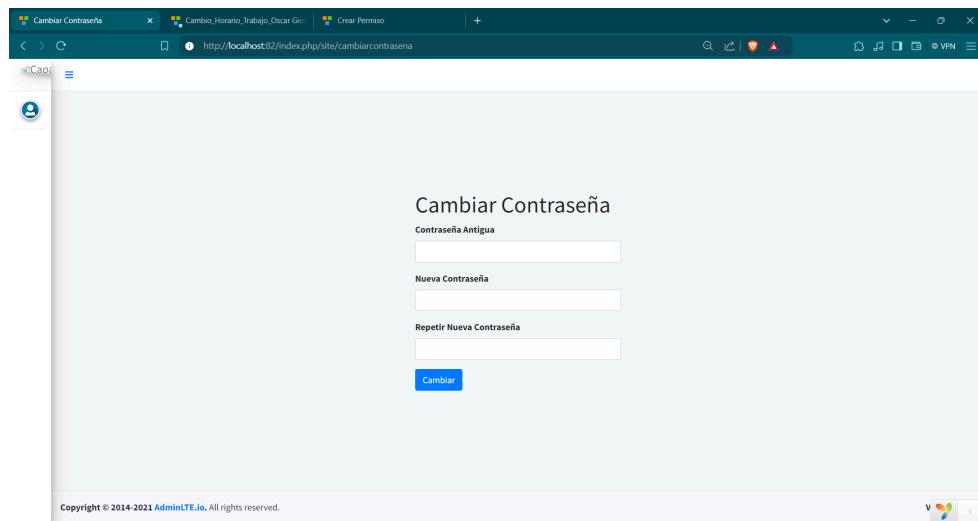
antecedente_ginecologico	p_menarca	int	Atributo para almacenar la edad a la que inició la menarca.
antecedente_ginecologico	p_menopausia	int	Atributo para almacenar la edad a la que inició la menopausia.
antecedente_ginecologico	p_f_u_m	date	Atributo para guardar la fecha de inicio de la última menstruación.
antecedente_ginecologico	p_f_u_citologia	date	Atributo para guardar la fecha de la última citología.
antecedente_ginecologico	p.Alteracion_frecuencia	varchar(25)	Atributo para guardar la respuesta seleccionada de la alteración de la menstruación.
antecedente_ginecologico	p.Alteracion_duracion	varchar(25)	Atributo para guardar la respuesta seleccionada de la alteración de la menstruación.
antecedente_ginecologico	p.Alteracion_cantidad	varchar(25)	Atributo para guardar la respuesta seleccionada de la alteración de la menstruación.
antecedente_ginecologico	p.Inicio_vida_s	int	Atributo para almacenar la edad del inicio de la vida sexual.
antecedente_ginecologico	p_no_parejas	int	Atributo para almacenar el número de parejas.
antecedente_ginecologico	p_vaginitis,p_cervicitis_mucopurulenta, p_chancroide, p_clamidia, p_eip, p_gonorrea, p_hepatitis, p_herpes, p_lgv, p_molusco_cont, p_ladillas, p_sarna, p_sifilis, p_tricomoniasis, p_vb, p_vih, p_vph	tinyint	Atributos para almacenar las respuestas seleccionadas (0 y 1) de enfermedades .
antecedente_ginecologico	observacion	text	Atributo para almacenar el texto de las observaciones de los antecedentes ginecológicos.

### Diseño de interfaz.

- Vista: Login y cambiar contraseña.

- Descripción: En esta ventana se encuentra el acceso al sistema mediante las credenciales de usuarios. En caso de ser un usuario nuevo se redirige a la ventana de cambiar contraseña.

A screenshot of the CAPASU login interface. It features a circular logo at the top with the word "CAPASU". Below it is a white rectangular form with the title "Inicia sesión". The form contains two input fields: "Usuario" and "Contraseña", each with a small icon next to it (a person for user and a lock for password). At the bottom of the form is a blue "Inicia sesión" button. The background is light gray.



- **Vista: Portal inicial (médico, empleado, gestor-rh).**
  - Descripción: En estas ventanas se encuentra el inicio de los usuarios en base a su rol.

The screenshot shows a web browser window titled 'Avisos' with a sub-tab 'Inicio'. The URL is 'http://localhost:82/index.php/site/portalgestionrh'. On the left, a sidebar menu includes icons for Home, Employees, and Notices. The main content area has two sections: 'Solicitudes Recientes' (Recent Requests) and 'Avisos' (Notices).  
**Solicitudes Recientes:**  
A table with columns 'Empleado', 'Fecha de Creación', and 'Acciones'. The data is as follows:

Empleado	Fecha de Creación	Acciones
Este Es Un Empleado	jueves, 07 de agosto de 2025	<button>Ver</button>
Este Es Un Empleado	jueves, 08 de agosto de 2024	<button>Ver</button>
Este Es Un Empleado	jueves, 08 de agosto de 2024	<button>Ver</button>
Este Es Un Empleado	jueves, 08 de agosto de 2024	<button>Ver</button>
Este Es Un Empleado	jueves, 08 de agosto de 2024	<button>Ver</button>

  
**Avisos:**  
A section titled 'Avisos' with a 'Crear Nuevo Aviso' button. It displays a single notice card:

**ESPACIO RECUPERADO**  
NO ARROJES BASURA  
EN ESTE LUGAR  
TU TAMBIÉN PUEDES  
APORTAR AL CUIDADO DEL  
MEDIO AMBIENTE

AVISO DE: 09 de agosto del 2024, 12:05 p.m.  
Origen: Instituto de Infraestructura y Construcción S.A.S

Inicio gestor-rh.

The screenshot shows a web browser window titled "PÁGINA DE INICIO- MEDICO" with the URL "localhost:82/index.php/site/portal-medico". The main content area is titled "Solicitudes de Citas Medicas Recientes" (Recent Medical Appointment Requests) and displays a table with the following data:

Empleado	Fecha de Creación	Acciones
Jesus Jose Alo Vera	martes, 06 de agosto de 2024 a las 12:11 horas	<button>Ver</button>
Oscar Giovanni Carillo Ibarrola	martes, 06 de agosto de 2024 a las 12:11 horas	<button>Ver</button>
Oscar Giovanni Carillo Ibarrola	martes, 06 de agosto de 2024 a las 12:11 horas	<button>Ver</button>
Empleado Exp Medico Prueba	martes, 06 de agosto de 2024 a las 12:11 horas	<button>Ver</button>
Oscar Giovanni Carillo Ibarrola	martes, 06 de agosto de 2024 a las 12:11 horas	<button>Ver</button>

At the bottom left, it says "Copyright © 2014-2021 AdminLTE.io. All rights reserved." and the bottom right has standard browser controls.

Inicio medico.

The screenshot shows a web browser window titled "Avisos" with the URL "localhost:82/index.php/site/portalempleado". The main content area shows a profile picture of a woman and the text "Este Es Un Empleado" and "Número de empleado: 527". Below this, there are tabs for "Información personal", "Información laboral", "Vacaciones", and "Solicitudes". Under "Información personal", there are fields for "No. Empleado", "Nombre", "Apellido", "CURP", "NSS", "Fecha Nacimiento", "Edad", "Sexo", "Estado civil", and "RFC". To the right, there is a sidebar titled "FORMATOS DE INCIDENCIAS" with a section for "Avisos" showing a thumbnail of a poster about environmental awareness.

Inicio empleado.

### ● Vista: Empleados.

- Descripción: En esta ventana se encuentra la vista general para el manejo de los empleados.

The screenshot shows a web browser window titled "Empleados" with the URL "localhost:82/index.php/empleados". The main content area is titled "LISTA DE EMPLEADOS" and shows a table with the following data:

#	Foto	Empleado	Número de empleado	Departamento	Dirección
1		Ad Min Caso Administracion	3278	N/A	2.- ADMINISTRACIÓN
2		Alfaro Huanost Martin	4	DEPARTAMENTO DE CATASTRO Y PADRÓN DE USUARIOS	3.- COMERCIAL
3		Alo Vera Jesus Jose	223	UNIDAD DE INFORMÁTICA	1.- GENERAL
4		Aviles Garcia Alberto	9	UNIDAD JURÍDICA	1.- GENERAL

At the top right, there is a button "AÑADIR NUEVO EMPLEADO". The bottom of the page shows a footer with the text "Mostrando 1-25 de 47 elementos."

- Formularios:

Formulario crear empleado.

- Vista: Ver empleado.

- Descripción: En esta ventana se encuentra la vista que muestra una serie de pestañas con toda la información del empleado (laboral, personal, de contacto, documentos, historial de solicitudes, expediente médico).

Pestaña información personal.

Días Laborales

Días laborales actuales:

Tipo de contrato \* Fecha de ingreso \* Departamento \*

Jefe inmediato Director de dirección Nombramiento \*

Número de cuenta Salario

Pestaña información laboral.

Total de días vacacionales: 26

**PRIMER PERÍODO**

JUEVES, 11 DE ABRIL DE 2024 — MARTES, 23 DE ABRIL DE 2024

Días de vacaciones disponibles: 13  
Días de vacaciones restantes: 0

Año: 2023  
Seleccionar rango de fechas del primer periodo:  
Original: No

**SEGUNDO PERÍODO**

DOMINGO, 01 DE DICIEMBRE DE 2024 — VIERNES, 13 DE DICIEMBRE DE 2024

Días de vacaciones disponibles: 13  
Días de vacaciones restantes: 0

Año: 2024  
Seleccionar rango de fechas del segundo periodo:  
Original: No

Pestaña información vacacional.

HISTORIAL DE SOLICITUDES DE INCIDENCIAS:

#	Empleado	Fecha Creacion	Tipo de solicitud
1	Samuel Perez Sanchez	miércoles, 31 de julio de 2024	CITA MEDICA
2	Samuel Perez Sanchez	miércoles, 31 de julio de 2024	CAMBIO DE PERÍODO VACACIONAL
3	Samuel Perez Sanchez	miércoles, 31 de julio de 2024	CAMBIO DE PERÍODO VACACIONAL
4	Samuel Perez Sanchez	miércoles, 31 de julio de 2024	CAMBIO DE PERÍODO VACACIONAL

Pestaña historial de solicitudes.

Enfermedad	Abuelos	Hermanos	Madre	Padre
Alergias	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Anomalías Congénitas	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Cáncer	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Cardiopatías	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Diabetes	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Daltonismo	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ceguera congénita	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Pestaña antecedentes hereditarios.

Pestaña antecedentes patológicos.

Pestaña antecedentes no patológicos.

**Antecedente Perinatal**

Hora de Nacimiento    No. de Gestación    Edad Gestacional

Site of birth

Peso al nacer    Talla

Perímetros (cm)

Cefálico	Torácico	Abdominal
2		

¿Cuenta con evaluaciones neonatales?

Pestaña antecedente perinatal.

**Antecedente Ginecológico**

Menarche    Menopausia    F.U.M    F.U. Cytología

Alteraciones de la menstruación

de Frecuencia	de Duración	de Cantidad
Seleccione si tiene alguna	Seleccione si tiene alguna	Seleccione si tiene alguna

Inicio vida sexual    Número de parejas

Vaginitis  
 Cervicitis Mucopurulenta  
 Chancroide  
 Clamidia

Pestaña antecedente ginecológico.

**Antecedente Obstétrico**

Fecha probable de parto.

Gestación    Aborto    Parto    Cesarea

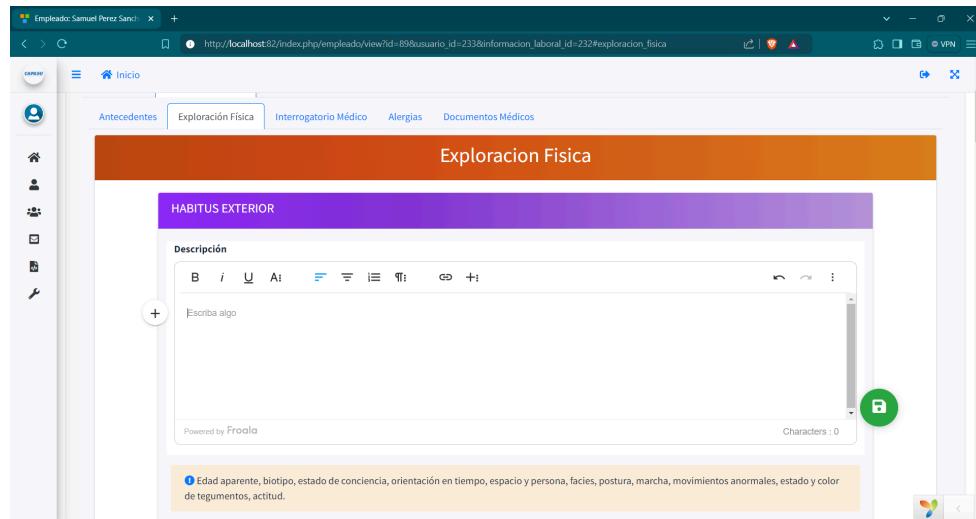
1	2		
---	---	--	--

Nacidos vivos    Viven

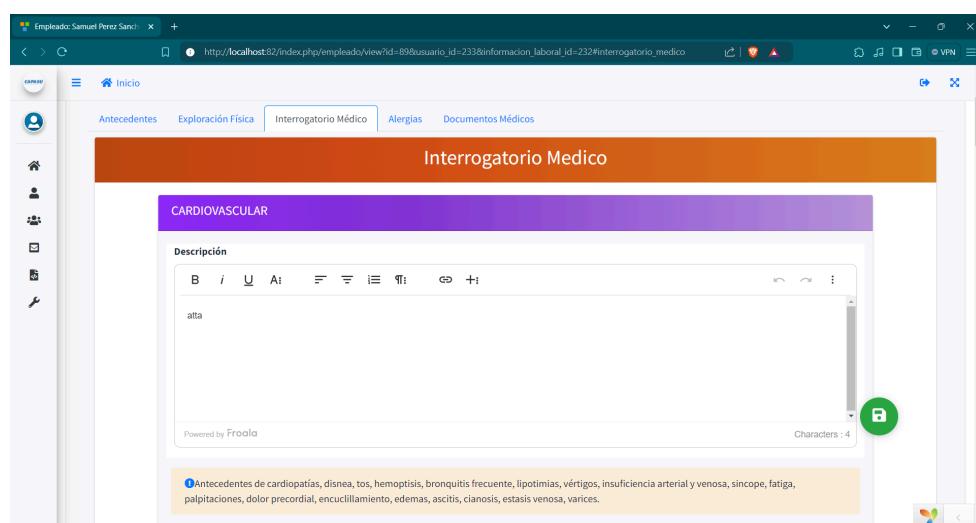
Nacidos Muertos    Muerto - 1ra semana  
Muerto después - 1ra semana

Intergenesia  
 Malformaciones  
 Atención prenatal

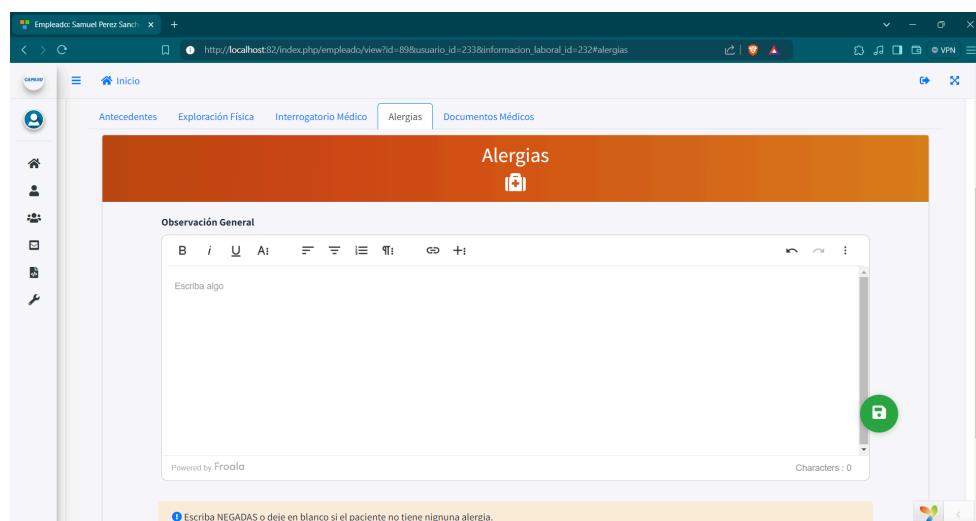
Pestaña antecedente obstétrico.



Pestaña exploración física.



Pestaña interrogatorio médico.



Pestaña alergias.

#	Fecha de la consulta	Motivo de la consulta
1	domingo, 11 de agosto de 2024	ss

Pestaña historial consultas medicas.

#	Nombre	Fecha Subida	Observación
1	estudio de	miércoles, 31 de julio de 2024	estudio de sangre

Pestaña documentos médicos.

#	Nombre	Fecha Subida	Observaciones
1	CURP	miércoles, 31 de julio de 2024	CURP

Pestaña documentos laborales y personales.

### - Formularios:

The screenshot shows a web-based document management system. At the top, there's a header bar with icons for user profile, search, and navigation. Below it is a sidebar with various icons. The main area has a teal header bar with the text 'Tipo de Documento'. Underneath, there's a dropdown menu labeled 'Seleccione el tipo de documento'. A rich text editor with toolbar buttons (B, i, U, AI, etc.) is present. Below the editor is a text input field with placeholder text 'Escriba algo' and a character count of 'Characters: 0'. A note at the bottom says 'Powered by Froala'. To the right of the editor is a 'Observaciones' section with a rich text editor and a text input field. Below these is a large 'Archivo' section with a placeholder 'Arrastre y suelte aquí los archivos ...'. At the bottom of this section are three buttons: 'Seleccionar archivo ...', 'Quitar', 'Examinar ...', and 'Subir archivo'. A yellow 'Cancelar' button is located in the top right corner.

Formulario documentos.

This screenshot shows a similar document management form. The header bar and sidebar are identical to the previous one. The main area has a teal header bar with the text 'Nombre del documento'. Below it is a text input field labeled 'Ingrese el nombre del documento'. A rich text editor with toolbar buttons follows. Below the editor is a text input field with placeholder text 'Escriba algo' and a character count of 'Characters: 0'. A note at the bottom says 'Powered by Froala'. To the right of the editor is an 'Observación' section with a rich text editor and a text input field. Below these is an 'Archivo' section with a placeholder 'Arrastre y suelte aquí los archivos ...'. At the bottom of this section are three buttons: 'Seleccionar archivo ...', 'Quitar', 'Examinar ...', and 'Subir archivo'. A yellow 'Cancelar' button is located in the top right corner.

Formulario documentos médicos.

This screenshot shows a medical consultation form. The header bar and sidebar are identical to the previous ones. The main area has a teal header bar with the text 'Consulta médica'. Below it is a text input field with placeholder text 'Expediente: Oscar Giovanni Cañito Isarrála'. The form is divided into sections: 'Interrogatorio' (with a rich text editor) and 'SINTOMAS' (with a rich text editor). Both sections have placeholder text 'Escriba algo' and a character count of 'Characters: 0'. A note at the bottom of each section says 'Powered by Froala'. A yellow 'Cancelar' button is located in the top right corner.

Formulario consulta médica.

### ● Vista: Historial general de solicitudes.

- Descripción: En esta ventana se encuentra la lista que muestra una lista de todas las solicitudes realizadas por los empleados.

#	Empleado	Fecha Creacion	Tipo de solicitud
1	Oscar Giovanni Carillo Ibarrola	jueves, 08 de agosto de 2024	PERMISO ECONOMICO
2	Este Es Un Empleado	jueves, 08 de agosto de 2024	PERMISO SIN GOCE DE SUELDO
3	Este Es Un Empleado	jueves, 08 de agosto de 2024	PERMISO SIN GOCE DE SUELDO
4	Este Es Un Empleado	jueves, 08 de agosto de 2024	PERMISO ECONOMICO
5	Este Es Un Empleado	jueves, 08 de agosto de 2024	PERMISO ECONOMICO
6	Este Es Un Empleado	jueves, 08 de agosto de 2024	PERMISO ECONOMICO
7	Este Es Un Empleado	jueves, 08 de agosto de 2024	CAMBIO DE HORARIO DE TRABAJO
8	Este Es Un Empleado	jueves, 08 de agosto de 2024	PERMISO FUERA DEL TRABAJO

### ● Vista: Ver solicitud.

- Descripción: En esta ventana se encuentra la información detallada de la solicitud realizada por el empleado.

Solicitud	
Número de empleado	650
Tipo de formato	PERMISO ECONOMICO
Nombre	Oscar Giovanni Carillo Ibarrola
Fecha Creacion	jueves, 08 de agosto de 2024
Motivo	siii
Fecha de Permiso	jueves, agosto 08, 2024

### ● Vista: Configuración.

- Descripción: En esta ventana se encuentran las pestañas que permiten realizar algunos ajustes del sistema.

#	Tipo de formato:	Tipo de Contrato:	Límite Anual
1	PERMISO FUERA DEL TRABAJO	Confianza	9
2	PERMISO FUERA DEL TRABAJO	Sindicalizado	10
3	COMISION ESPECIAL	Sindicalizado	14
4	COMISION ESPECIAL	Confianza	15
5	CAMBIO DE DIA LABORAL	Confianza	8
6	CAMBIO DE DIA LABORAL	Sindicalizado	9
7	CAMBIO DE HORARIO DE TRABAJO	Sindicalizado	5

Pestaña ajustes formatos.

#	Nombramiento:
1	OFICIAL DE OPERACIONES ESPECIAL
2	INSPECTOR ESPECIAL
3	JEFE DE CONTROL VEHICULAR
4	JEFE DEL DEPARTAMENTO DE RECURSOS Y SERVICIOS GENERALES
5	JEFE DE ATENCION A USUARIOS
6	DIRECTORA DE ADMINISTRACION
7	DIRECCION DE CONTABILIDAD

Pestaña ajustes nombramientos.

#	Departamento:	Dirección	DPTO
1	DIRECCIÓN GENERAL	1.- GENERAL	DGG
2	COMISARÍA	1.- GENERAL	DGUC
3	UNIDAD DE CONTRALORÍA	1.- GENERAL	DGCI
4	UNIDAD DE DIFUSIÓN Y CULTURA DEL AGUA	1.- GENERAL	DGUD
5	UNIDAD DE INFORMÁTICA	1.- GENERAL	DGUI
6	UNIDAD JURÍDICA	1.- GENERAL	DGUJ
7	UNIDAD DE MANTENIMIENTOS FISICOS	1.- GENERAL	DGMF

Pestaña ajustes departamentos.

### - Formularios:

**Límites de formatos**

Formatos:

Tipo de Contrato:

Límite Anual:

**Guardar**

Formulario límite de formatos.

**Nombramientos:**

Nombre Puesto:

**Guardar**

Formulario nombramiento.

**Departamento:**

Nombre Departamento:

Dirección:

DPTO:

**Guardar**

Formulario departamento.

- Vista: Permisos (administración de usuarios, roles, permisos, asignaciones).

- Descripción: En esta ventana se encuentra el menú administrativo del control de usuarios.

#	Nombre del Empleado	Usuario	Email	Status
1	Peña Paniagua Sergio Eli	speñap2	(no definido)	Active
2	Perez E Nose Jose	npereze	(no definido)	Active
3	Mero Mero El Director	emerom	(no definido)	Active
4	Humanos Si Usuario Recursos	uhumanoss	(no definido)	Active
5	Ad Min Caso Administracion	cadm	(no definido)	Active
6	El Director Director Admin	deld	(no definido)	Active
7	De Departamento Jefe Admin	jded	(no definido)	Active
8	Empleado Otro Otro Administrador	oempleadoo	(no definido)	Active

Pestaña lista usuarios.

#	Nombre del Empleado	Usuario	
1	Peña Paniagua Sergio Eli	speñap2	⊕
2	Perez E Nose Jose	npereze	⊕
3	Mero Mero El Director	emerom	⊕
4	Humanos Si Usuario Recursos	uhumanoss	⊕
5	Ad Min Caso Administracion	cadm	⊕
6	El Director Director Admin	deld	⊕
7	De Departamento Jefe Admin	jded	⊕
8	Empleado Otro Otro Administrador	oempleadoo	⊕
9	Uno Nuevo Prueba De	punon	⊕
10	Sindi Prueba Prueba Sindi	psindip	⊕
11	Prueba E Otro Vacaciones	opruebae	⊕

Pestaña asignaciones.

**Roles**

Mostrando 1-4 de 4 elementos.

#	Nombre	Nombre de regla	Descripción	Acciones
1	administrador	(no definido)	Acceso total a todo el sistema	
2	empleado	(no definido)	Acceso a funciones principales solo de empleados	
3	gestor-rh	(no definido)	acceso a funciones de gestor de recursos humanos	
4	medico	(no definido)	Acceso a funciones de medico	

**Asignaciones**

- Usuarios
- Asignaciones
- Roles
- Permisos
- Rutas
- Reglas
- Menús

**Pestaña roles.**

**Permisos**

Mostrando 1-20 de 29 elementos.

#	Nombre	Nombre de regla	Descripción	Acciones
1	acceso-formatos-incidencias	(no definido)	El usuario tiene permitido acceder a los formularios de los formatos de incidencias	
2	acciones-documentos-medicos	(no definido)	El usuario tiene permitido agregar, eliminar, descargar los documentos médicos del empleado	
3	borrar-registros-formatos-incidencias	(no definido)	El usuario tiene permitido eliminar los registros de los formatos de incidencias. (Permiso fuera del trabajo, comisión especial, cambio de día labora, permiso económico, etc.)	
4	Control-total	(no definido)	El usuario tiene control total del sistema (Acceso a todas las rutas y todos los permisos)	

**Asignaciones**

- Usuarios
- Asignaciones
- Roles
- Permisos
- Rutas
- Reglas
- Menús

**Pestaña permisos.**

**Rutas**

Nueva Ruta(s)

Buscar Disponible

- /admin/assignment/index
- /admin/assignment/view
- /admin/assignment/assign
- /admin/assignment/revoke
- /admin/assignment/\*
- /admin/default/index
- /admin/default/\*
- /admin/menu/index
- /admin/menu/view
- /admin/menu/create
- /admin/menu/update
- /admin/menu/delete
- /admin/menu/\*
- /admin/permission/index
- /admin/permission/view
- /admin/permission/create
- /admin/permission/update
- /admin/permission/delete
- /admin/permission/assign
- /admin/permission/get-users

Buscar Asignado

- /\*
- /cambio-dia-laboral/\*
- /cambio-dia-laboral/delete
- /cambio-dia-laboral/export-html
- /cambio-dia-laboral/export-html-se
- /cambio-dia-labora/historial
- /cambio-dia-labora/view
- /cambio-horario-trabajo/\*
- /cambio-horario-trabajo/delete
- /cambio-horario-trabajo/export-htm
- /cambio-horario-trabajo/export-htm
- /cambio-horario-trabajo/historial
- /cambio-horario-trabajo/view
- /cambio-periodo-vacacional/\*
- /cambio-periodo-vacacional/delete
- /cambio-periodo-vacacional/export
- /cambio-periodo-vacacional/export
- /cambio-periodo-vacacional/historial
- /cambio-periodo-vacacional/view
- /cita-medica/\*

**Asignaciones**

- Usuarios
- Asignaciones
- Roles
- Permisos
- Rutas
- Reglas
- Menús

**Pestaña rutas.**

### - Formularios:

Crear Rol

Nombre:

Descripción:

Nombre de regla:

Datos:

Crear

Powered by Yii Framework

Usuarios  
Asignaciones  
**Roles**  
Permisos  
Rutas  
Reglas  
Menús

< Volver

Formulario crear rol.

Crear Permiso

Nombre:

Descripción:

Nombre de regla:

Datos:

Crear

Powered by Yii Framework

Usuarios  
Asignaciones  
Roles  
**Permisos**  
Rutas  
Reglas  
Menús

< Volver

Formulario crear permiso.

### ● Vista: Incidencias.

- Descripción: Ventanas de los diferentes tipos de incidencias (historial, vista detallada, formulario, impresión).
- Historial:

The screenshot shows a web-based application window titled "Citas Medicas". The main title bar says "Historial de Citas Medicas". Below it, there's a button "Crear una nueva" (Create a new one) and a "Inicio" (Home) button. A message at the top indicates "Mostrando 1-23 de 23 elementos". A table lists 23 entries, each with a number (#), date of creation (Fecha de creación), and reason (Motivo). All entries show the same reason: "malestar". Each row has a small "eye" icon to its right.

#	Fecha de creación	Motivo
1	2024-07-29	malestar
2	2024-07-29	malestar
3	2024-07-29	malestar
4	2024-07-29	malestar
5	2024-07-29	malestar
6	2024-07-29	malestar
7	2024-07-29	malestar
8	2024-07-29	malestar
9	2024-07-29	malestar

*Historial citas medicas..*

The screenshot shows a web-based application window titled "Permiso Fuerza del Trabajo". The main title bar says "Historial de Permisos fuera de trabajo". Below it, there's a button "Crear una nueva" (Create a new one) and a "Inicio" (Home) button. A message at the top indicates "Mostrando 1-11 de 11 elementos". A table lists 11 entries, each with a number (#), date of creation (Fecha de creación), and reason (Motivo). The reasons are varied and include "abundancia", "uno", "UN MOTIVO", "un motivo", "MMMMMMMMMMMMMM", "MMMMMMMMMMMMMM", "unnnnoooo", and "Un motivo equis tal asi y asi, paso esto y asi". Each row has a small "eye" icon to its right.

#	Fecha de creación	Motivo
1	2024-08-08	abundancia
2	2024-08-08	uno
3	2025-08-07	UN MOTIVO
4	2024-07-24	un motivo
5	2024-07-23	MMMMMMMMMMMMMM
6	2024-07-23	MMMMMMMMMMMMMM
7	2024-07-23	unnnnoooo
8	2024-07-23	Un motivo equis tal asi y asi, paso esto y asi

*Historial permisos fuera de trabajo.*

The screenshot shows a web-based application window titled "Comisión Especial". The main title bar says "Historial de Comision Especial". Below it, there's a button "Crear una nueva" (Create a new one) and a "Inicio" (Home) button. A message at the top indicates "Mostrando 1-15 de 15 elementos". A table lists 15 entries, each with a number (#), date of creation (Fecha de creación), and reason (Motivo). The reasons are "COMISION ALLA", "listo", "unooooo", "motivoooooooooooooooo", "Motivo tataata", and "un motivo". Each row has a small "eye" icon to its right.

#	Fecha de creación	Motivo
1	2024-08-08	COMISION ALLA
2	2024-08-08	listo
3	2024-07-30	unooooo
4	2024-07-24	motivoooooooooooooooo
5	2024-07-24	Motivo tataata
6	2024-07-24	un motivo

*Historial comisión especial.*

The screenshot shows a web-based application interface for managing workday changes. The title bar reads "Cambio de Día Laboral". The main content area is titled "Historial de Cambio de Día Laboral". It displays a table with columns: #, Fecha de creación, and Motivo. The table contains 9 rows of data, each with an "eye" icon for viewing details. The data includes various dates from July 24 to August 8, 2024, and reasons such as "TAL", "motivoooooo taatataatata", and "un motivo". A large bold text "un motivo" is centered below the table.

#	Fecha de creación	Motivo
1	2024-08-08	TAL
2	2024-08-08	TAL
3	2024-08-08	TAL
4	2024-07-24	motivoooooo taatataatata
5	2024-07-24	<b>un motivo</b>
6	2024-07-24	un motivo
7	2024-07-23	Lorem ipsum dolor sit amet consectetur adipisicing elit. Sit laborum dolorum veritatis adipisci fuga ex. rem velit

*Historial cambio de día laboral.*

The screenshot shows a web-based application interface for managing work hour changes. The title bar reads "Cambio de Horario de Trabajo". The main content area is titled "Historial de Cambio de Horario de Trabajo". It displays a table with columns: #, Fecha de creación, and Motivo. The table contains 5 rows of data, each with an "eye" icon for viewing details. The data includes dates from July 16 to August 8, 2024, and reasons such as "UNOS MOTIVOS", "porque siiiii", "un motivo", "Lorem ipsum dolor sit amet consectetur adipisicing elit. Totam nulla ut, alias provident minima saepe suscipit quo placeat aperiam? Ex impedit inventore molestiae ad voluptatem provident mollitia nihil officia quos!", and "TEMPORAL".

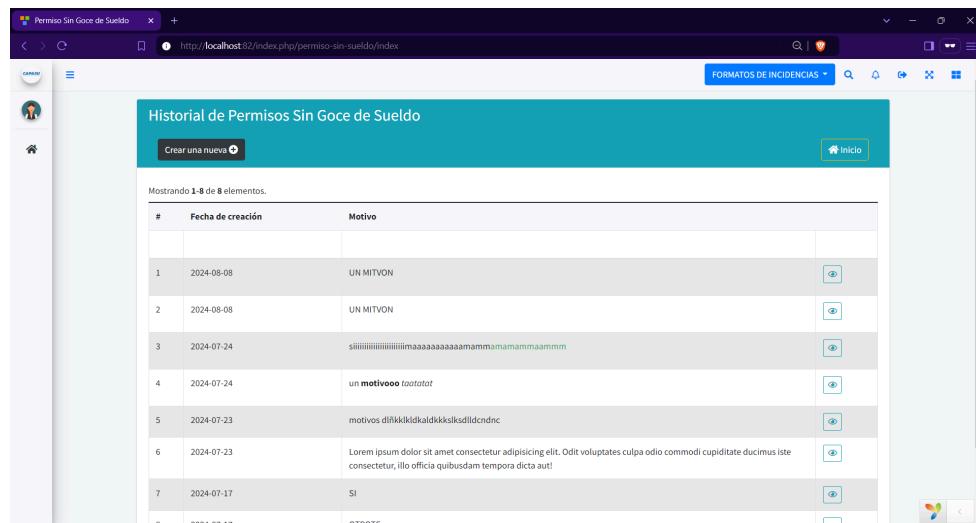
#	Fecha de creación	Motivo
1	2024-08-08	UNOS MOTIVOS
2	2024-07-24	porque siiiii
3	2024-07-24	un motivo
4	2024-07-23	Lorem ipsum dolor sit amet consectetur adipisicing elit. Totam nulla ut, alias provident minima saepe suscipit quo placeat aperiam? Ex impedit inventore molestiae ad voluptatem provident mollitia nihil officia quos!
5	2024-07-16	TEMPORAL

*Historial cambio de horario de trabajo.*

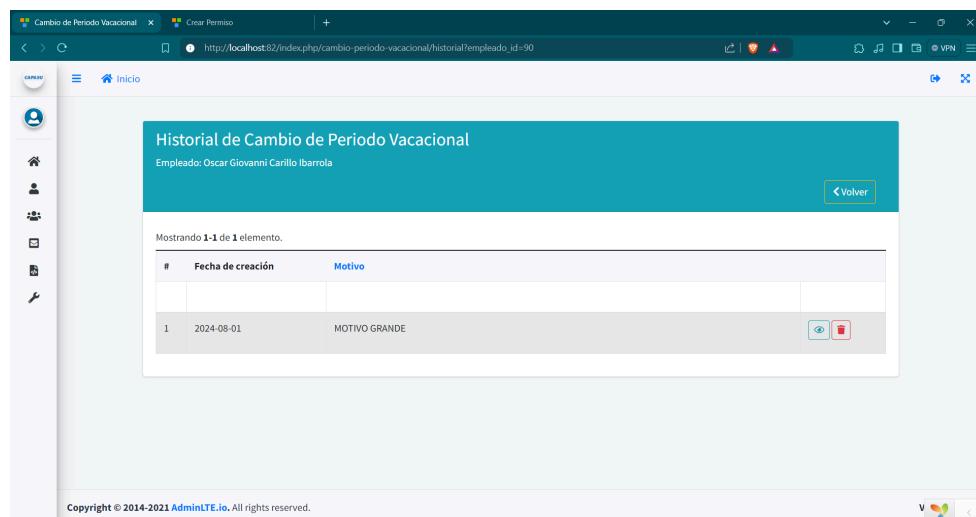
The screenshot shows a web-based application interface for managing economic leave. The title bar reads "Permiso Económico". The main content area is titled "Historial de Permisos Económicos". It displays a table with columns: #, Fecha de creación, and Motivo. The table contains 11 rows of data, each with an "eye" icon for viewing details. The data includes dates from July 24 to August 8, 2024, and reasons such as "sisis", "Permito die", "siiiiiiiiiiii oyooootttt mddmmdmdmdm", "Un motivo el siguiente totatot mmmmmmmmm", and "Un motivo".

#	Fecha de creación	Motivo
1	2024-08-08	sisis
2	2024-08-08	sisis
3	2024-08-08	sisis
4	2024-07-31	Permito die
5	2024-07-24	siiiiiiiiiiii oyooootttt mddmmdmdmdm
6	2024-07-24	<i>Un motivo el siguiente totatot mmmmmmmmm</i>
7	2024-07-24	<i>Un motivo</i>

*Historial permisos economicos.*

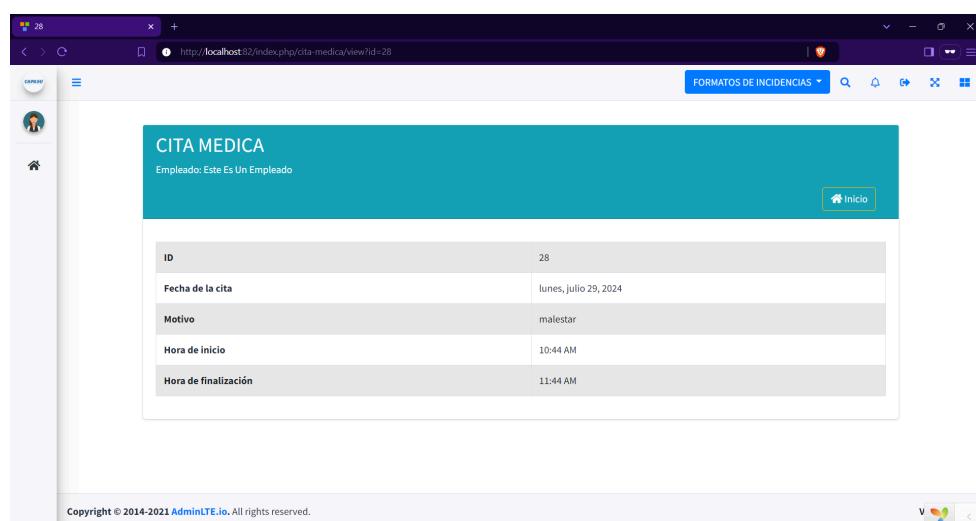


### *Historial permisos sin goce de sueldo.*



### *Historial cambio de periodo vacacional*

#### - Vista detalles:



Vista de cita médica.

The screenshot shows a software application window titled "PERMISO FUERA TRABAJO". The page header includes the text "Empleado: Este Es Un Empleado" and a "Vista Previa de Impresión" button. The main content area displays the following information in a table:

Solicitud ID	398
Motivo	abundancia
Hora de Salida	11:23 AM
Hora de Regreso	12:23 PM
Fecha que Repondrá	viernes, agosto 09, 2024
Horario de fecha que Repondrá	12:25 PM
Fecha de Permiso	jueves, agosto 08, 2024

At the bottom of the page, there is a copyright notice: "Copyright © 2014-2021 AdminLTE.io. All rights reserved."

Vista de permiso fuera de trabajo.

The screenshot shows a software application window titled "COMISIÓN ESPECIAL". The page header includes the text "Empleado: Este Es Un Empleado" and a "Vista Previa de Impresión" button. The main content area displays the following information in a table:

ID de Solicitud	366
Motivo	COMISION ALLA
Fecha de Comisión	domingo, agosto 11, 2024

At the bottom of the page, there is a copyright notice: "Copyright © 2014-2021 AdminLTE.io. All rights reserved."

Vista de comisión especial.

The screenshot shows a software application window titled "CAMBIO DE DÍA LABORAL". The page header includes the text "Empleado: Este Es Un Empleado" and a "Vista Previa de Impresión" button. The main content area displays the following information in a table:

Solicitud ID	372
Motivo	TAL
Día Que Laborará	viernes, agosto 09, 2024
Día que descansará	jueves, agosto 08, 2024

At the bottom of the page, there is a copyright notice: "Copyright © 2014-2021 AdminLTE.io. All rights reserved."

Vista de cambio de día laboral.

**CAMBIO DE HORARIO DE TRABAJO**

Empleado: Este Es Un Empleado

Vista Previa de Impresión Inicio

Solicitud ID	407
Motivo del cambio de horario	UNOS MOTIVOS
Fecha	viernes, agosto 09, 2024
Turno	MATUTINO
Inicio de horario	7:42 AM
Fin de horario	3:42 PM
Fecha de Inicio	viernes, agosto 09, 2024
Fecha de Termino	jueves, agosto 15, 2024

Vista de cambio de horario de trabajo.

**PERMISO ECONOMICO**

Empleado: Este Es Un Empleado

Vista Previa de Impresión Inicio

Solicitud ID	415
Motivo	sisis
Fecha de Permiso	sábado, agosto 10, 2024
Fecha de Permiso Anterior	jueves, agosto 08, 2024
No. de Permiso Anterior	10

Vista de permiso económico.

**PERMISO SIN GOCE DE SUELDO**

Empleado: Este Es Un Empleado

Vista Previa de Impresión Inicio

Solicitud ID	418
Motivo	UN MITVON
Fecha de Permiso	domingo, agosto 11, 2024
Fecha de Permiso Anterior	viernes, agosto 09, 2024
No. de Permiso Anterior	8

Vista de permiso sin goce de sueldo.

ID de Solicitud	314
Motivo	MOTIVO GRANDE
Año	2024
Periodo	1ero
Primera vez	Sí
Fecha de Inicio	martes, abril 02, 2024
Fecha de Fin	viernes, abril 12, 2024
Comentario	

Vista de cambio de periodo vacacional.

### - Formulario:

Fecha para la cita	Hora de inicio	Hora de fin
dd/mm/aaaa		

Motivo:

Escriba algo

Formulario cita médica.

Formulario permiso fuera de trabajo.

Formulario comisión especial.

Formulario cambio de día laboral.

Permisos usados: 5  
Permisos disponibles: 1

Fecha de permiso: dd/mm/aaaa Turno: Seleccione el Turno

Inicio de horario Termino de horario Rango de Fechas

Motivo:

Escriba algo

Powered by Froala

Formulario cambio de horario de trabajo.

Permisos usados: 11  
Permisos disponibles: 0

Has alcanzado el límite anual de permisos.

No Permiso Anterior  
11

Fecha Permiso Anterior  
2024-08-10

Fecha de permiso  
dd/mm/aaaa

Motivo:

Escriba algo

Formulario permiso económico.

Permisos usados: 8  
Permisos disponibles: 1

No Permiso Anterior  
9

Fecha Permiso Anterior  
2024-08-11

Fecha de permiso  
dd/mm/aaaa

Motivo:

Escriba algo

Formulario permiso sin goce de sueldo.

### *Formulario cambio de periodo vacacional.*

- Vista previa de impresión:

The screenshot shows a Microsoft Edge browser window with the following details:

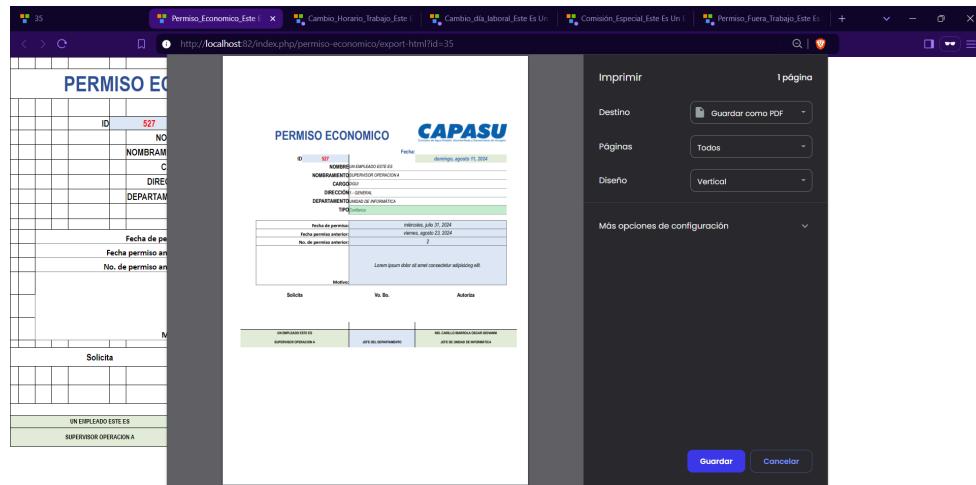
- Title Bar:** Microsoft Edge - Permisos, Fueras, Trabajo, Este Es...
- Address Bar:** http://localhost:82/index.php/permiso-fuera-trabajo/export-html?id=36
- Content Area:** A form titled "PERMISO FUERA DEL TRABAJO" from "CAPASU".
  - Employee Information:** ID: 557, NOMBRE: JOSÉ MIGUEL GONZALEZ ORTEGA, NOMBRE APELLIDO: GONZALEZ ORTEGA, DIRECCION: CALLE 123, DEPARTAMENTO: 101, TIPO: FAMILIAR.
  - Leave Details:** Fecha de permiso: 2024-07-15, Salida: 2024-07-15, Regreso: 2024-07-16.
  - Motivo:** Motivo: Vacaciones.
  - Reason:** Fecha que repondrá: 2024-07-15, Nota: No.
  - Solicita:** Solicita: No.
  - Notes:** UN EMPLEADO ESTE EN SUPERVISOR OPERACIONES A.
- Right Sidebar:** A modal window for printing.
  - Print Options:** Imprimir (Print), 1 página (One page).
  - Destino:** Guardar como PDF (Save as PDF), Todos (All).
  - Diseño:** Vertical.
  - More Options:** Más opciones de configuración (More configuration options).
- Buttons:** Guardar (Save), Cancelar (Cancel).

*Impresión formato permiso fuera de trabajo.*

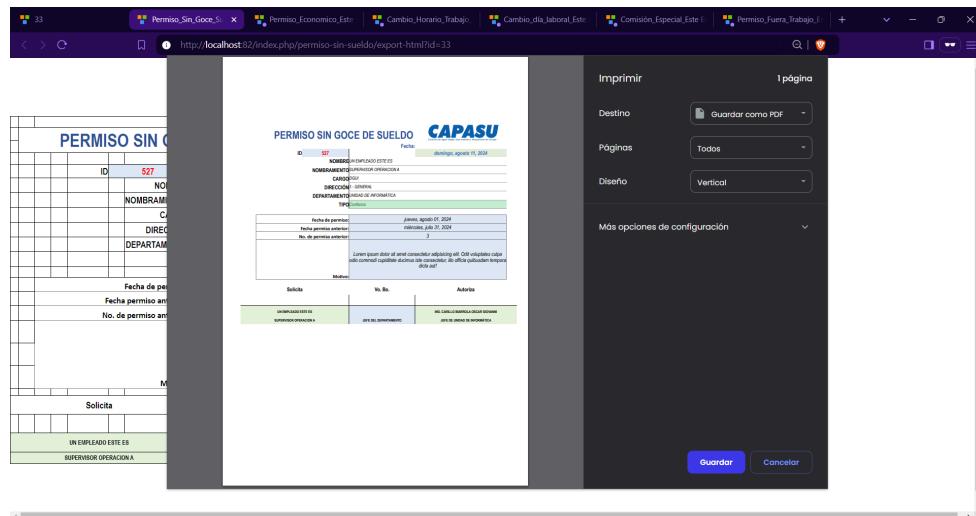
Impresión formato comisión especial.

Impresión formato cambio de día laboral.

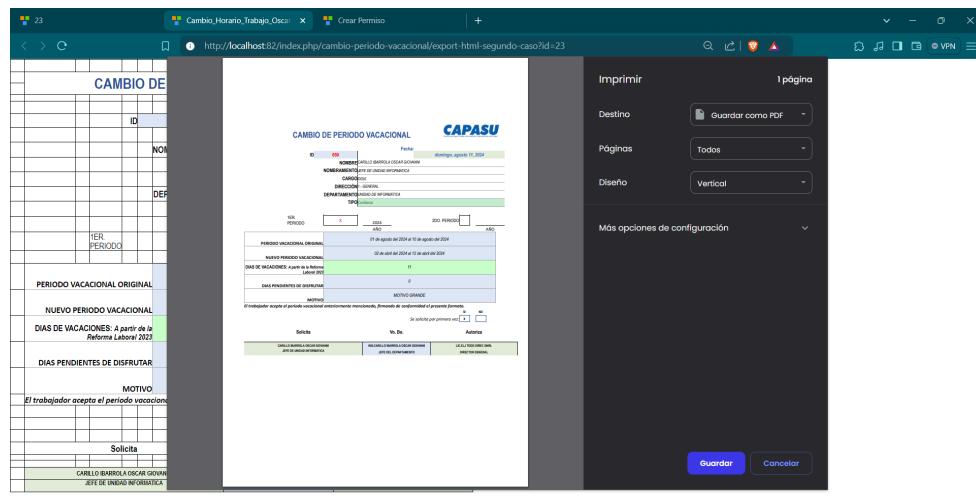
Impresión formato cambio de horario de trabajo.



Impresión formato de permiso económico.



Impresión formato de permiso sin goce de sueldo.



Impresión de formato de cambio de periodo vacacional.

## Desarrollo y personalización.

### Vistas.

#### Site.

- login.

Descripción. Código de la vista con la que interactúan todos los usuarios para poder acceder al sistema.

```
<?php
//CLASES IMPORTADAS
use yii\helpers\Html;
use yii\bootstrap4\ActiveForm;
use yii\bootstrap5\Alert;

?>
<div class="card ">
    <div class="card-body bg-light login-card-body">
        <p class="login-box-msg">Inicia sesión</p>

        <?php $form = ActiveForm::begin(['id' => 'login-form']) ?>
        <div class="d-flex align-items-center mb-3">

            <?php
            //ALERTAS
            foreach (Yii::$app->session->getAllFlashes() as $type =>
$messag...
                if ($type === 'error') {
                    echo Alert::widget([
                        'options' => ['class' => 'alert-danger'],
                        'body' => $message,
                    ]);
                } else {
                    echo Alert::widget([
                        'options' => ['class' => 'alert-' . $type],
                        'body' => $message,
                    ]);
                }
            }
        }
    }
}>
```

```
?>
</div>

<?=

//FORMULARIO DEL ACTIVEFORM
$form->field($model, 'username', [
    'options' => ['class' => 'form-group has-feedback'],
    'inputTemplate' => '{input}<div class="input-group-append"><div
class="input-group-text"><span class="fa fa-user" style="color:
#007bff;"></span></div></div>',
    'template' => '{beginWrapper}{input}{error}{endWrapper}',
    'wrapperOptions' => ['class' => 'input-group mb-3']
])
->label(false)
->textInput(['placeholder' => $model->getAttributeLabel('username')])?

?>

<?= $form->field($model, 'password', [
    'options' => ['class' => 'form-group has-feedback'],
    'inputTemplate' => '{input}<div class="input-group-append"><div
class="input-group-text"><span class="fas fa-lock" style="color:
#007bff;"></span></div></div>',
    'template' => '{beginWrapper}{input}{error}{endWrapper}',
    'wrapperOptions' => ['class' => 'input-group mb-3']
])
->label(false)
->passwordInput(['placeholder' =>
$model->getAttributeLabel('password')]) ?>

<div class="row">
    <div class="col-6">

        </div>
        <div class="col-6">
            <?= Html::submitButton('Inicia sesión', ['class' => 'btn btn-primary
btn-block']) ?>
        </div>
    </div>

    <?php ActiveForm::end(); ?>

</div>
```

</div>

- portalempleado.

Descripción. Código de la vista principal (Home) con la que interactúan todos los usuarios de rol “empleado ” cuando acceden al sistema. Vista también conocida como “Mi perfil” para usuarios con rol distinto al de empleado.

```
<?php
//IMPORTACIONES
use yii\web\View;
use kartiktabs\TabsX;
use hails12\adminlte\widgets\Alert;
use yii\widgets\DetailView;
use yii\helpers\Html;

$this->registerCssFile('@web/css/site.css', ['position' => View::POS_HEAD]);

$this->title = 'PAGINA DE INICIO - EMPLEADO';
//SE CARGAN LOS MODELOS
$activeTab = Yii::$app->request->get('tab', 'info_p');
$currentTime = date('Y-m-d');
$antecedentesExistentes = [];
$observacionGeneral = "";
$descripcionAntecedentes = "";
$modelAntecedenteNoPatologico = new
\app\models\AntecedenteNoPatologico();
$modelExploracionFisica = new \app\models\ExploracionFisica();
//PERMISO
$editable = Yii::$app->user->can('editar-expediente-medico');

if ($antecedentes) {
    foreach ($antecedentes as $antecedente) {

$antecedentesExistentes[$antecedente->cat_antecedente_hereditario_id][$a
ntecedente->parentezco] = true;
        if (empty($observacionGeneral)) {
            $observacionGeneral = $antecedente->observacion;
        }
    }
}
```

}

//SE OBTIENEN LOS REGISTROS

```
$modelAntecedentePatologico =  
\app\models\AntecedentePatologico::findOne(['expediente_medico_id' =>  
$expedienteMedico->id]);  
if (!$modelAntecedentePatologico) {  
    $modelAntecedentePatologico = new  
\app\models\AntecedentePatologico();  
    $modelAntecedentePatologico->expediente_medico_id =  
$expedienteMedico->id;  
}
```

```
$modelAntecedenteNoPatologico =  
\app\models\AntecedenteNoPatologico::findOne(['expediente_medico_id' =>  
$expedienteMedico->id]);  
if (!$modelAntecedenteNoPatologico) {  
    $modelAntecedenteNoPatologico = new  
\app\models\AntecedenteNoPatologico();  
    $modelAntecedenteNoPatologico->expediente_medico_id =  
$expedienteMedico->id;  
}
```

```
$modelExploracionFisica =  
\app\models\ExploracionFisica::findOne(['expediente_medico_id' =>  
$expedienteMedico->id]);  
if (!$modelExploracionFisica) {  
    $modelExploracionFisica = new \app\models\ExploracionFisica();  
    $modelExploracionFisica->expediente_medico_id =  
$expedienteMedico->id;  
}
```

```
$modellInterrogatorioMedico =  
\app\models\InterrogatorioMedico::findOne(['expediente_medico_id' =>  
$expedienteMedico->id]);  
if (!$modellInterrogatorioMedico) {  
    $modellInterrogatorioMedico = new \app\models\InterrogatorioMedico();  
    $modellInterrogatorioMedico->expediente_medico_id =  
$expedienteMedico->id;  
}
```

```
$modelAntecedentePerinatal =
\app\models\AntecedentePerinatal::findOne(['expediente_medico_id' =>
$expedienteMedico->id]);
if (!$modelAntecedentePerinatal) {
    $modelAntecedentePerinatal = new \app\models\AntecedentePerinatal();
    $modelAntecedentePerinatal->expediente_medico_id =
$expedienteMedico->id;
}

$modelAntecedenteGinecologico =
\app\models\AntecedenteGinecologico::findOne(['expediente_medico_id' =>
$expedienteMedico->id]);
if (!$modelAntecedenteGinecologico) {
    $modelAntecedenteGinecologico = new
\app\models\AntecedenteGinecologico();
    $modelAntecedenteGinecologico->expediente_medico_id =
$expedienteMedico->id;
}

$modelAntecedenteObstrectico =
\app\models\AntecedenteObstrectico::findOne(['expediente_medico_id' =>
$expedienteMedico->id]);
if (!$modelAntecedenteObstrectico) {
    $modelAntecedenteObstrectico = new
\app\models\AntecedenteObstrectico();
    $modelAntecedenteObstrectico->expediente_medico_id =
$expedienteMedico->id;
}

$modelAlergia = \app\models\Alergia::findOne(['expediente_medico_id' =>
$expedienteMedico->id]);
if (!$modelAlergia) {
    $modelAlergia = new \app\models\Alergia();
    $modelAlergia->expediente_medico_id = $expedienteMedico->id;
}

?>
<div class="container-fluid">
    <div class="row ">
        <?php
```

```
//ALERTA
foreach (Yii::$app->session->getAllFlashes() as $type => $message) {
    if ($type === 'error') {
        echo Alert::widget([
            'options' => ['class' => 'alert-danger'],
            'body' => $message,
        ]);
    } else {
        echo Alert::widget([
            'options' => ['class' => 'alert-' . $type],
            'body' => $message,
        ]);
    }
}
?>

<div class="col-md-8">
    <?=
        //SE RENDERIZA LA VISTA QUE SE ENCUENTRA EN
        EMPLEADO Y SE CARGAN LOS MODELOS
        //CORRESPONDIENTES AL EMPLEADO
        $this->render('//empleado/view', [
            'model' => $model,
            'documentos' => $documentos,
            'documentoModel' => $documentoModel,
            'historial' => $historial,
            'searchModelConsultas' => $searchModelConsultas,
            'dataProviderConsultas' => $dataProviderConsultas,
            'searchModel' => $searchModel,
            'dataProvider' => $dataProvider,
            'expedienteMedico' => $expedienteMedico,
            'antecedentes' => $antecedentes,
            'catAntecedentes' => $catAntecedentes,
            'antecedenteNoPatologico' => $antecedenteNoPatologico,
            'ExploracionFisica' => $ExploracionFisica,
            'InterrogatorioMedico' => $InterrogatorioMedico,
            'AntecedentePerinatal' => $AntecedentePerinatal,
            'AntecedenteGinecologico' => $AntecedenteGinecologico,
            'AntecedenteObstetrico' => $AntecedenteObstetrico,
            'Alergia' => $Alergia,
```

```
'antecedentePatologico' => $antecedentePatologico,
'documentoMedicoModel' => $documentoMedicoModel
]) ?>

</div>

<div class="col-md-4">
<div class="card bg-light">
<div class="card-header bg-info text-white">
<h4>Avisos</h4>
</div>
<div class="card-body">

<div style="max-height: 700px; overflow-y: auto;">

<?php setlocale(LC_TIME, "es_419.UTF-8"); ?>

<?php
//SE RENDERIZA LA VISTA DE CARRUSEL-AVISOS DE
AVISO, Y SE
//CARGA EL MODELO DE AVISO
echo $this->render('//aviso/carrusel-avisos', [
    'avisos' => $avisos,
]);
?>
</div>
</div>
</div>
</div>

</div>

</div>
```

- portalgestionrh.

Descripción. Código de la vista principal (Home) con la que interactúan todos los usuarios de rol “gestion-rh ” cuando acceden al sistema.

```
<?php
```

```
//IMPORTACIONES
use yii\bootstrap5\Alert;
use yii\helpers\Html;

use yii\web\View;
//CARGAR ESTILOS CSS
$this->registerCssFile('@web/css/site.css', ['position' => View::POS_HEAD]);

$this->title = 'PAGINA DE INICIO- ADMIN';

?>
<div class="container-fluid">
    <div class="row ">
        <?php

//ALERTA
foreach (Yii::$app->session->getAllFlashes() as $type => $message) {
    if ($type === 'error') {
        echo Alert::widget([
            'options' => ['class' => 'alert-danger'],
            'body' => $message,
        ]);
    } else {
        echo Alert::widget([
            'options' => ['class' => 'alert-' . $type],
            'body' => $message,
        ]);
    }
}
?>

<div class="col-md-8">
    <div class="card">

        <div class="card-header gradient-yellow text-white">
            <h2> Solicitudes de Citas Medicas Recientes <i class="fa fa-heartbeat"></i></h2>
        </div>

        <div class="card-body">
            <div class="row">
                <div class="col-sm-12">

                    <div style="max-height: 300px; overflow-y: auto;">
```

```
<?php setlocale(LC_TIME, "es_419.UTF-8"); ?>
<table class="table table-bordered table-hover">
    <thead class="thead-dark">
        <tr>
            <th>Empleado</th>

            <th>Fecha de Creación</th>
            <th>Acciones</th>
        </tr>
    </thead>
    <tbody>
        <?php // SE GENERA LA TABLA CON LA SOLICITUDES MÁS
RECIENTES REALIZADAS
        foreach ($solicitudesRecientes as $solicitud): ?>
        <tr>
            <td>
                <?= $solicitud->empleado ?
Html::encode($solicitud->empleado->nombre . ' ' . $solicitud->empleado->apellido) : 'N/A' ?>
            </td>

            <td><?= strftime("%A, %d de %B de %Y",
strptime($solicitud->fecha_creacion)) ?></td>
            <td>
                <?=

                    Html::a('Ver', ['solicitud/view', 'id' => $solicitud->id], ['class'
=> 'btn btn-primary btn-sm']) ?>
            </td>
        </tr>
    <?php endforeach; ?>
    </tbody>
</table>

</div>

</div>
</div>
```

```
</div>

<div class="col-md-4">
    <div class="card bg-light">

        <div class="card-header bg-info text-white">
            <h4>Avisos</h4>
        </div>

        <div class="card-body">

            <?php setlocale(LC_TIME, "es_419.UTF-8"); ?>

            <?php

                //SE RENDERIZA LA VISTA DE CARRUSEL-AVISOS DE
                AVISO, Y SE
                //CARGA EL MODELO DE AVISO
                echo $this->render('//aviso/carrusel-avisos', [
                    'avisos' => $avisos,

                ]);
                ?>
            </div>
        </div>
    </div>

</div>

</div>
```

- portalmedico.

Descripción. Código de la vista principal (Home) con la que interactúan todos los usuarios de rol “gestion-rh ” cuando acceden al sistema.

```
<?php
//IMPORTACIONES
use yii\helpers\Html;
use hait812\adminlte\widgets\Alert;
```

```
use yii\web\View;
$this->registerCssFile('@web/css/site.css', ['position' => View::POS_HEAD]);

$this->title = 'PAGINA DE INICIO- MEDICO';
$this->params['breadcrumbs'] = [['label' => $this->title]];
?>
<div class="container-fluid">
<div id="dynamic-content">

</div>

<div class="container-fluid">
<div class="row justify-content-center">
<div class="col-sm-7">
<div class="card">

<div class="card-header gradient-yellow text-white">
<h2> Solicitudes de Citas Medicas Recientes <i class="fa fa-heartbeat"></i></h2>
</div>

<div class="card-body">
<div class="row">
<div class="col-sm-12">

<div style="max-height: 300px; overflow-y: auto;">
<?php setlocale(LC_TIME, "es_419.UTF-8");?>
<table class="table table-bordered table-hover">
<thead class="thead-dark">
<tr>
<th>Empleado</th>

<th>Fecha de Creación</th>
<th>Acciones</th>
</tr>
</thead>
<tbody>
<?php
// SE GENERA LA TABLA CON LA SOLICITUDES DE CITAS
MEDICAS MÁS RECIENTES REALIZADAS
foreach ($solicitudesRecientes as $solicitud): ?>
<tr>
<td>
```

```
<?= $solicitud->empleado ?
Html::encode($solicitud->empleado->nombre . '' .
$solicitud->empleado->apellido) : 'N/A' ?>
</td>

<td>
<?= strftime('%A, %d de %B de %Y', strtotime($solicitud->created_at)) ?>
a las <?= date('H:i', strtotime($solicitud->created_at)) ?> horas
</td>
<td>
<?= Html::a('Ver', ['solicitud/view', 'id' => $solicitud->id],
['class' => 'btn btn-primary btn-sm']) ?>
</td>
</tr>
<?php endforeach; ?>
</tbody>
</table>
</div>
</div>
</div>

</div>
</div>
</div>
</div>
</div>

</div>
```

- error.

Descripción. Código que muestra algún tipo de error que se produzca en el sistema (página no encontrada, acceso denegado, etc).

```
<?php

/* @var $this yii\web\View */
/* @var $name string */
/* @var $message string */
/* @var $exception Exception */

use yii\helpers\Html;
```

```
$this->title = $name;
$this->params['breadcrumbs'] = [['label' => $this->title]];
?>
<div class="error-page">
    <div class="error-content" style="margin-left: auto;">
        <h3><i class="fas fa-exclamation-triangle text-danger"></i> <?= Html::encode($name) ?></h3>

        <p>
            <?=
                //SE OBTIENE EL MENSAJE DEL ERROR PRODUCIDO EN EL
                SISTEMA
                nl2br(Html::encode($message)) ?>
            </p>

        <p>
            El error anterior ocurrió mientras el servidor web estaba procesando su
            solicitud. Comuníquese con nosotros si cree que se trata de un error del
            servidor. Gracias. Mientras tanto, puedes<a
            href="javascript:history.back(-1);">
                regresar.
        </p>
    </div>
```

- cambiarcontrasena.

Descripción. Código que muestra el formulario con el que el usuario puede interactuar para cambiar su contraseña.

```
<?php
//IMPORTACIONES
use yii\helpers\Html;
use yii\widgets\ActiveForm;

/* @var $this yii\web\View */
/* @var $model app\models\CambiarContrasenaForm */
/* @var $form ActiveForm */

$this->title = 'Cambiar Contraseña';
$this->params['breadcrumbs'][] = $this->title;
?>
<div class="site-cambiarcontrasena">
    <h1><?= Html::encode($this->title) ?></h1>

    <?php
    // FORMULARIO PARA EL CAMBIO DE CONTRASEÑA DEL USUARIO
```

```
$form = ActiveForm::begin(); ?>

<?= $form->field($model, 'oldPassword')->passwordInput() ?>
<?= $form->field($model, 'newPassword')->passwordInput() ?>
<?= $form->field($model, 'repeatNewPassword')->passwordInput() ?>

<div class="form-group">
    <?= Html::submitButton('Cambiar', ['class' => 'btn btn-primary']) ?>
</div>
<?php ActiveForm::end(); ?>
</div>
```

- configuración.

Descripción. Código que renderiza las vistas (indexes), de cat-puesto, cat-departamento, parametro-formato.

```
<?php
//IMPORTACIONES
use kartik\tabs\TabsX;

use yii\helpers\Html;

use yii\grid\GridView;
use yii\widgets\Pjax;

use yii\jui\DatePicker;

?>

<div class="container-fluid">

<?php
//BLOQUES PARA AÑADIRLOS AL TABS
$this->beginBlock('block-parametro-formato'); ?>
<?php
// REDENRIZAR VISTA Y CARGAR MODELO DE
PARAMETRO_FORMATO
echo $this->render('//parametro-formato/index', [
    'searchModel' => $parametroFormatoSearchModel,
    'dataProvider' => $parametroFormatoDataProvider,
]);
?>

<?php $this->endBlock();?>
```

```
<?php $this->beginBlock('block-departamentos'); ?>
<?php
    // REDENRIZAR VISTA Y CARGAR MODELO DE
    CAT_DEPARTAMENTO
    echo $this->render('//cat-departamento/index', [
        'searchModel' => $catDepartamentoSearchModel,
        'dataProvider' => $catDepartamentoDataProvider
    ]);
    ?>
<?php $this->endBlock();?>

<?php $this->beginBlock('block-puestos'); ?>
<?php
    // REDENRIZAR VISTA Y CARGAR MODELO DE CAT_PUESTO
    echo $this->render('//cat-puesto/index', [
        'searchModel' => $catPuestoSearchModel,
        'dataProvider' => $catPuestoDataProvider,
    ]);
    ?>
<?php $this->endBlock();?>

<?php
//ORGANIZAR PESTAÑAS
$tabs = [
    [
        'label' => 'Formatos',
        'content' => $this->blocks['block-parametro-formato'],
        'active' => true,
        'options' => [
            'id' => 'conf-formatos',
        ],
    ],
    [
        'label' => 'Nombramientos',
        'content' => $this->blocks['block-puestos'],
        // 'active' => true,
        'options' => [
            'id' => 'conf-puestos',
        ],
    ],
    [
        'label' => 'Departamentos',
        'content' => $this->blocks['block-departamentos'],
        // 'active' => true,
        'options' => [

```

```
'id' => 'conf-departamentos',
],
],
];
//MOSTRAR LOS BLOQUES EN LOS TABS
echo TabsX::widget([
    'enableStickyTabs' => true,
    'options' => ['class' => 'custom-tabs-2'],
    'items' => $tabs,
    'position' => TabsX::POS_ABOVE,
    'align' => TabsX::ALIGN_CENTER,
    'encodeLabels' => false,
]);
?>
</div>
```

### Empleado.

- form.  
Descripción. Código que muestra el formulario con el que se interactúa para crear el registro de un empleado.

```
<?php
//IMPORTACIONES
use app\models\CatDepartamento;
use app\models\CatDireccion;
use app\models\CatPuesto;
use app\models\CatTipoContrato;
use yii\helpers\Html;
use yii\bootstrap4\ActiveForm;
use kartik\file\FileInput;
use app\models\Departamento;
use kartik\select2\Select2;
use yii\bootstrap5\Alert;
use yii\helpers\ArrayHelper;
use yii\web\View;
/* @var $this yii\web\View */
/* @var $model app\models\Empleado */
/* @var $form yii\bootstrap4\ActiveForm */
//CARGAR ESTILOS
$this->registerCssFile('@web/css/site.css', ['position' => View::POS_HEAD]);
```

```
$this->registerCssFile('@web/css/grid-view.css', ['position' =>
View::POS_HEAD]);
?>

<div class="container-fluid">
    <div class="row justify-content-center">
        <div class="col-md-10">
            <div class="card">
                <?php
                    //FORMULARIO DE CREACION DE EMPLEADO
                    $form = ActiveForm::begin(['options' => ['enctype' =>
'multipart/form-data', 'id' => 'form-empleado']]);
                ?>
                <div id="loading-spinner-laboral" style="display: none;">
                    <i class="fa fa-spinner fa-spin fa-2x" style="color: #000000;"></i>
                Procesando...
                </div>
                <div class="card-header bg-primary text-white">
                    <h2>AÑadir Nuevo Empleado</h2>
                    <?= Html::submitButton('Guardar y enviar datos de usuario <i
class="fa fa-save fa-md"></i>', [
                        'class' => 'btn btn-light btn-lg float-right',
                        'id' => 'save-button-personal'
                    ]) ?>
                </div>
                <div class="card-body">
                    <div class="row">
                        <div class="col-md-12">
                            <div class="d-flex align-items-center mb-3">
                                <?php
                                    //ALERTA
                                    foreach (Yii::$app->session->getAllFlashes() as $type =>
$message) {
                                        if ($type === 'error') {
                                            echo Alert::widget([
                                                'options' => ['class' => 'alert-danger'],
                                                'body' => $message,
                                            ]);
                                        } else {
                                            echo Alert::widget([
                                                'options' => ['class' => 'alert-' . $type],
                                                'body' => $message,
                                            ]);
                                        }
                                    }
                                </div>
                            </div>
                        </div>
                    </div>
                </div>
            </div>
        </div>
    </div>
</div>
```

```
    }
    ?>
</div>

<div class="empleado-form">
    <div class="row">
        <div class="col-md-6">
            <div class="card">
                <div class="card-header bg-info text-white">Información del Empleado</div>
                <div class="card-body">
                    <div class="col-6 col-sm-8">
                        <?= $form->field($usuario,
'rol')->dropDownList([
                            1 => 'Trabajador',
                            2 => 'Gestor de recursos humanos',
                            3 => 'Medico'
                        ], ['prompt' => 'Selecciona el
rol'])->label('Seleccione el rol del empleado:') ?>
                    </div>
                    <div class="col-6 col-sm-6">
                        <?= $form->field($model,
'numero_empleado')->input('number', ['maxlength' => 4, 'class' =>
"form-control"]) ?>
                    </div>
                    <div class="col-6 col-sm-10">
                        <?= $form->field($model,
'nombre')->textInput(['maxlength' => true])->label('Nombre del empleado:') ?>
                    </div>
                    <div class="col-6 col-sm-10">
                        <?= $form->field($model,
'apellido')->textInput(['maxlength' => true])->label('Apellido del empleado:') ?>
                    </div>
                    <div class="col-6 col-sm-10">
                        <?= $form->field($model,
'email')->textInput(['maxlength' => true])->label('Email del empleado:') ?>
                    </div>
                    <?=
                        //SE CARGA LA EXTENSIÓN DE KARTIK /
FILEINPUT PARA SUBIR ARCHIVOS
                        $form->field($model,
'foto')->widget(FileInput::classname(), [

```

```
// 'options' => ['accept' => 'file/*'],
  'pluginOptions' => [
    'showUpload' => false,
    'showCancel' => false,
    'allowedFileExtensions' => ['jpeg', 'jpg',
      'png'],
      ],
    ]) -> label('Foto del empleado:') ?>
  </div>
</div>
</div>
<div class="col-md-6">
  <div class="card">
    <div class="card-header gradient-verde
text-white">Información Laboral</div>
  <div class="card-body">
    <div class="col-6 col-sm-12">
      <?=
        // SE CARGA EL WIDGET (EXTENSION) SELECT2 CON LOS DATOS
        DEL MODELO DE CAT_DEPARTAMENTO
        $form->field($informacion_laboral,
          'cat_departamento_id')->widget(Select2::classname(), [
            'data' => ArrayHelper::map(CatDepartamento::find()->all(), 'id',
              'nombre_departamento'),
            'language' => 'es',
            'options' => ['placeholder' => 'Seleccione departamento'],
            'pluginOptions' => ['allowClear' => true],
            'theme' => Select2::THEME_BOOTSTRAP,
          ]) -> label('Departamento al que pertenece:') ?>
      </div>

      <div class="col-6 col-sm-12">
        <?=
          // SE CARGA EL WIDGET (EXTENSION) SELECT2 CON LOS DATOS
          DEL MODELO DE CAT_PUESTO
          $form->field($informacion_laboral,
            'cat_puesto_id')->widget(Select2::classname(), [
              'data' => ArrayHelper::map(CatPuesto::find()->all(), 'id',
                'nombre_puesto'),
              'language' => 'es',
              'options' => ['placeholder' => 'Seleccione puesto del empleado'],
              'pluginOptions' => ['allowClear' => true],
              'theme' => Select2::THEME_BOOTSTRAP,
            ]) -> label('Puesto del empleado:') ?>
```

```

</div>

<div class="col-6 col-sm-8">
    <?=
        //SE CARGA EL WIDGET (EXTENSION) SELECT2 CON LOS DATOS
        DEL MODELO DE CAT_TIPO_CONTRATO
        $form->field($informacion_laboral,
        cat_tipo_contrato_id)->widget(Select2::classname(), [
            'data' => ArrayHelper::map(CatTipoContrato::find()->all(), 'id',
            nombre_tipo),
            'language' => 'es',
            'options' => ['placeholder' => 'Seleccione el tipo de contrato'],
            'pluginOptions' => ['allowClear' => true],
            'theme' => Select2::THEME_BOOTSTRAP,
        ])->label('Tipo de contrato del empleado:') ?>
</div>

<div class="col-6 col-sm-8">
    <?= $form->field($model, 'profesion')->dropDownList([
        'No tiene' => 'No tiene',
        'ING.' => 'ING.',
        'LIC.' => 'LIC.',
        'PROF.' => 'PROF.',
        'ARQ.' => 'ARQ.',
        'C.' => 'C.',
        'DR.' => 'DR.',
        'DRA.' => 'DRA.',
        'TEC.' => 'TEC.',
    ], ['prompt' => 'Selecciona el nivel académico...'])->label('Profesión:') ?>
</div>

<div class="col-6 col-sm-6">
    <?= $form->field($informacion_laboral,
    fecha_ingreso)->input('date')->label('Fecha de ingreso del empleado:') ?>
</div>
</div>

                </div>
            </div>
        </div>
    </div>
</div>
</div>

<?php ActiveForm::end(); ?>

```

```
<?php
//SESCRIPT PARA ALERTAR AL USUARIO QUE SU SOLICITUD
SE ESTRA PROCESANDO
$script = <<< JS
$('#form-empleado').on('beforeSubmit', function() {
var button = $('#save-button-personal');
var spinner = $('#loading-spinner-laboral');

button.prop('disabled', true); // Deshabilita el botón
spinner.show(); // Muestra el spinner

return true; // Permite que el formulario se envíe
});
JS;
$this->registerJs($script);
?>
</div>
</div>
</div>
</div>
```

- create.

Descripción. Código que renderiza la vista \_form y carga los modelos utilizados en los formularios.

```
<?php

use yii\helpers\Html;

/* @var $this yii\web\View */
/* @var $model app\models\Empleado */

$this->title = 'Create Empleado';
$this->params['breadcrumbs'][] = ['label' => 'Empleados', 'url' => ['index']];
$this->params['breadcrumbs'][] = $this->title;
?>

<div class="container-fluid">

<div class="row">
<div class="col-md-12">
<?=
// RENDERIZA EL _FORM Y CARGA LOS MODELOS DE
USUARIO, INFORMACION_LABORAL
```

```
// Y JUNTA_GOBIERNO
$this->render('_form', [
    'model' => $model,
    'usuario' => $usuario,
    'informacion_laboral' => $informacion_laboral,
    'juntaGobiernoModel' => $juntaGobiernoModel

]) ?>
</div>
</div>

</div>

- index.

Descripción. Código que muestra la lista de los registros de empleados. En esta vista se definen permisos para que los usuarios con algún rol o asignación en específico tengan accesibles ciertas acciones de control de empleados.

<?php
// IMPORTACIONES
use yii\helpers\Html;
use yii\grid\GridView;
use yii\widgets\Pjax;
use hain812\adminlte3\yii\grid\ActionColumn;
use yii\web\View;
use kartik\tabs\TabsX;
use yii\bootstrap4\ActiveForm;
use yii\helpers\ArrayHelper;
use app\models\CatDireccion;
use kartik\select2\Select2;
use app\models\CatDepartamento;
use app\models\Empleado;
use yii\helpers\Url;
use app\models\JuntaGobiernoSearch;
use yii\bootstrap5\Alert;
/* @var $this yii\web\View */
/* @var $searchModel app\models\EmpleadoSearch */
/* @var $dataProvider yii\data\ActiveDataProvider */

//CARGAR ESTILOS
$this->registerCssFile('@web/css/site.css', ['position' => View::POS_HEAD]);

$this->title = 'Empleados';
$this->params['breadcrumbs'][] = $this->title;
```

```
\yii\web\YiiAsset::register($this);
//PERMISO
$visible = Yii::$app->user->can('medico');
?>
<div class="container-fluid">
<div class="row justify-content-center">
<div class="col-md-12">

    <div class="row justify-content-center">
    <div class="col-md-10">
        <div class="d-flex align-items-center mb-3 float-right">
            <?php
//ALERTA

foreach (Yii::$app->session->getAllFlashes() as $type =>
$message) {
    if ($type === 'error') {
        echo Alert::widget([
            'options' => ['class' => 'alert-danger'],
            'body' => $message,
        ]);
    } else {
        echo Alert::widget([
            'options' => ['class' => 'alert-' . $type],
            'body' => $message,
        ]);
    }
}
?>
</div>

<?php $this->beginBlock('block-empleados'); ?>
<div class="card">
    <div class="card-header bg-info text-white">
        <h3>LISTA DE EMPLEADOS</h3>

        <?php
//PERMISO PARA IR A CREATE EMPLEADO
if (Yii::$app->user->can('manejo-empleados')): ?>
            <?= Html::a('AÑADIR NUEVO EMPLEADO <i class="fa fa-user-plus fa-lg"></i>', ['create'], ['class' => 'btn btn-warning float-right fa-lg btn-dark']) ?>
```

```
<?php endif; ?>

</div>

<div class="card-body">
    <?php Pjax::begin(); ?>
    <?php
        //PERMISO PARA MOSTRAR LA LISTA COMPLETA DE TODOS LOS
        REGISTROS DE EMPLEADO
        if (Yii::$app->user->can('ver-todos-empleados')) { ?>
            <?= GridView::widget([
                'dataProvider' => $dataProvider,
                'filterModel' => $searchModel,
                'columns' => [
                    ['class' => 'yii\grid\SerialColumn'],
                    [
                        'class' => 'yii\grid\DataColumn',
                        'header' => '1ra. Revisión Medica',
                        'format' => 'raw',
                        'visible' => $visible,
                        'value' => function ($model) {
                            // Verificar si expedienteMedico no es null
                            if ($model->expedienteMedico !== null) {
                                if ($model->expedienteMedico->primera_revision === 0) {
                                    return '<i class="fas fa-exclamation-triangle" style="color: #ea4242 "></i> <br> PENDIENTE';
                                } elseif ($model->expedienteMedico->primera_revision === 1) {
                                    return '<i class="fas fa-check-circle" aria-hidden="true" style="color: #4678fc"></i>';
                                } else {
                                    return '<i class="fa fa-question" aria-hidden="true" style="color: #6c757d"></i>';
                                }
                            } else {
                                return '<i class="fa fa-question" aria-hidden="true" style="color: #6c757d"></i>'; // Valor predeterminado si expedienteMedico es null
                            }
                        },
                    ],
                    'contentOptions' => ['class' => 'text-center'],
                    'headerOptions' => ['class' => 'text-center'],
                ],
            ]);
        }
    
```

```
[  
    'attribute' => 'foto',  
    'format' => 'html',  
    'filter' => false,  
    'value' => function ($model) {  
        if ($model->foto) {  
            $urlImagen =  
                Yii::$app->urlManager->createUrl(['empleado/foto-empleado', 'id' =>  
                    $model->id]);  
            return Html::img($urlImagen, ['width' => '80px', 'height' =>  
                '80px']);  
        }  
        return null;  
    },  
,  
[  
    'attribute' => 'id',  
    'label' => 'Empleado',  
    'value' => function ($model) {  
        return $model ? $model->apellido . ' ' . $model->nombre :  
            'N/A';  
    },  
    'filter' => Select2::widget([  
        'model' => $searchModel,  
        'attribute' => 'id',  
        'data' =>  
            \yii\helpers\ArrayHelper::map(\app\models\Empleado::find()->all(), 'id',  
            function ($model) {  
                return $model->apellido . ' ' . $model->nombre;  
            }),  
        'options' => ['placeholder' => 'Empleado'],  
        'pluginOptions' => [  
            'allowClear' => true  
        ],  
        'theme' => Select2::THEME_KRAJEE_BS3,  
    ]),  
    'contentOptions' => ['class' => 'small-font'],  
,  
[  
    'attribute' => 'numero_empleado',  
    'label' => 'Número de empleado',  
    'value' => function ($model) {  
        return $model->numero_empleado;  
    },  
    'filter' => Select2::widget([
```

```
'model' => $searchModel,
'attribute' => 'numero_empleado',
'data' =>
\yii\helpers\ArrayHelper::map(\app\models\Empleado::find()->select(['numero
_empleado'])->distinct()->all(), 'numero_empleado', 'numero_empleado'),
'options' => ['placeholder' => 'Número Empleado'],
'pluginOptions' => [
    'allowClear' => true
],
'theme' => Select2::THEME_KRAJEE_BS3,
]),
'contentOptions' => ['class' => 'small-font'],
],
[
    'attribute' => 'cat_departamento_id',
    'label' => 'Departamento',
    'value' => function ($model) {
        return $model->informacionLaboral &&
$model->informacionLaboral->catDepartamento
    ?
$model->informacionLaboral->catDepartamento->nombre_departamento
: 'N/A';
},
'filter' => Select2::widget([
    'model' => $searchModel,
    'attribute' => 'cat_departamento_id',
    'data' => ArrayHelper::map(CatDepartamento::find()->all(), 'id',
'nombre_departamento'),
    'options' => ['placeholder' => 'Departamento', 'class' =>
'small-select2'],
    'pluginOptions' => [
        'allowClear' => true,
    ],
    'theme' => Select2::THEME_KRAJEE_BS3,
]),
'contentOptions' => ['class' => 'small-font'],
'headerOptions' => ['class' => 'small-font'],
],
[
    'attribute' => 'cat_direccion_id',
    'label' => 'Dirección',
    'value' => function ($model) {
        return $model->informacionLaboral &&
$model->informacionLaboral->catDireccion
    ?
$model->informacionLaboral->catDireccion->nombre_direccion
```

```
: 'N/A';
},
'filter' => Select2::widget([
    'model' => $searchModel,
    'attribute' => 'cat_direccion_id',
    'data' => ArrayHelper::map(CatDireccion::find()->all(), 'id',
        'nombre_direccion'),
    'options' => ['placeholder' => 'Dirección'],
    'pluginOptions' => [
        'allowClear' => true,
    ],
    'theme' => Select2::THEME_KRAJEE_BS3,
]),
'contentOptions' => ['class' => 'small-font'],
],
[
    'class' => ActionColumn::class,
    // if (Yii::$app->user->can('btn-vista-crear-empleado')):

    'template' => Yii::$app->user->can('manejo-empleados') ? '{view}
{delete} {toggle-activation}' : '{view}',
    'buttons' => [
        'toggle-activation' => function ($url, $model) {
            $isActive = $model->usuario->status == 10;
            $icon = $isActive ? 'fas fa-ban' : 'far fa-check-circle';
            $title = $isActive ? 'Desactivar Usuario' : 'Activar Usuario';
            return Html::a('<i class="' . $icon . '"></i>',
['empleado/toggle-activation', 'id' => $model->id], [
                'title' => Yii::t('yii', $title),
                'data-confirm' => Yii::t('yii', '¿Estás seguro de que deseas
cambiar el estado de este usuario?'),
                'data-method' => 'post',
                'class' => 'btn btn-xs ' . ($isActive ? 'btn-warning' :
'btn-success'),
            ]);
        },
        'view' => function ($url, $model) {
            return Html::a('<i class="far fa-eye"></i>', $url, [
                'title' => 'Ver información de empleado',
                'class' => 'btn btn-info btn-xs',
                'data-pjax' => "0"
            ]);
        },
        'delete' => function ($url, $model) {
            return Html::a('<i class="fas fa-trash"></i>', $url, [
                'title' => Yii::t('yii', 'Eliminar'),
            ]);
        },
    ],
]);
```

```
'data-confirm' => Yii::t('yii', '¿Estás seguro de que deseas
eliminar este elemento?'),
'data-method' => 'post',
'class' => 'btn btn-danger btn-xs',
]);
},
],
],
],
'summaryOptions' => ['class' => 'summary mb-2'],
'pager' => [
    'class' => 'yii\bootstrap4\LinkPager',
],
]);
}); ?>
```

```
<?php
//MOSTRAR LISTA DE EMPLEADOS POR DIRECCION O
DEPARTAMENTO
} elseif (Yii::$app->user->can('ver-empleados-direccion') ||
Yii::$app->user->can('ver-empleados-departamento')) { ?>
<?= GridView::widget([
    'dataProvider' => $dataProvider,
    'filterModel' => $searchModel,
    'columns' => [
        'class' => 'yii\grid\SerialColumn',
        [
            'attribute' => 'foto',
            'format' => 'html',
            'filter' => false,
            'value' => function ($model) {
                if ($model->foto) {
                    $urlImagen =
                        Yii::$app->urlManager->createUrl(['empleado/foto-empleado', 'id' =>
$model->id]);
                    return Html::img($urlImagen, ['width' => '80px', 'height' =>
'80px']);
                }
                return null;
            },
        ],
        [
            'attribute' => 'id',
            'label' => 'Empleado',
            'value' => function ($model) {
                return $model ? $model->apellido . ' ' . $model->nombre : 'N/A';
            }
        ]
    ]
]); ?>
```

```
},
'filter' => Select2::widget([
    'model' => $searchModel,
    'attribute' => 'id',
    'data' => $nombresEmpleadosData,
    'options' => ['placeholder' => 'Empleado'],
    'pluginOptions' => [
        'allowClear' => true
    ],
    'theme' => Select2::THEME_KRAJEE_BS3,
]),
'contentOptions' => ['class' => 'small-font'],
],  
  
[
    'attribute' => 'numero_empleado',
    'label' => 'Número de empleado',
    'value' => function ($model) {
        return $model->numero_empleado;
    },
    'filter' => Select2::widget([
        'model' => $searchModel,
        'attribute' => 'numero_empleado',
        'data' => $numeroEmpleadosData,
        'options' => ['placeholder' => 'Número Empleado'],
        'pluginOptions' => [
            'allowClear' => true
        ],
        'theme' => Select2::THEME_KRAJEE_BS3,
]),
    'contentOptions' => ['class' => 'small-font'],
],
[
    'attribute' => 'cat_departamento_id',
    'label' => 'Departamento',
    'value' => function ($model) {
        return $model->informacionLaboral &&
$model->informacionLaboral->catDepartamento
    ?
$model->informacionLaboral->catDepartamento->nombre_departamento
        : 'N/A';
    },
    'filter' => Select2::widget([
        'model' => $searchModel,
        'attribute' => 'cat_departamento_id',
        'data' => $departamentosData, // Usa los datos filtrados
```

```
'options' => ['placeholder' => 'Departamento', 'class' =>
'small-select2'],
'pluginOptions' => [
  'allowClear' => true,
],
'theme' => Select2::THEME_KRAJEE_BS3,
]),
'contentOptions' => ['class' => 'small-font'],
'headerOptions' => ['class' => 'small-font'],
],
[

  'class' => 'yii\grid\ActionColumn',

  'template' => Yii::$app->user->can('manejo-empleados') ? '{view}
{delete} {toggle-activation}' : '{view}',

  'buttons' => [
    'toggle-activation' => function ($url, $model) {
      $isActive = $model->usuario->status == 10;
      $icon = $isActive ? 'fas fa-ban' : 'far fa-check-circle';
      $title = $isActive ? 'Desactivar Usuario' : 'Activar Usuario';
      return Html::a('<i class="' . $icon . '"></i>',
['empleado/toggle-activation', 'id' => $model->id], [
      'title' => Yii::t('yii', $title),
      'data-confirm' => Yii::t('yii', '¿Estás seguro de que deseas
cambiar el estado de este usuario?'),
      'data-method' => 'post',
      'class' => 'btn btn-xs ' . ($isActive ? 'btn-warning' :
'btn-success'),
    ]);
  },
  'view' => function ($url, $model) {
    return Html::a('<i class="far fa-eye"></i>', $url, [
      'title' => 'Ver información de empleado',
      'class' => 'btn btn-outline-info btn-sm',
      'data-pjax' => "0"
    ]);
  },
  'delete' => function ($url, $model) {
    return Html::a('<i class="fas fa-trash"></i>', $url, [
      'title' => Yii::t('yii', 'Eliminar'),
      'data-confirm' => Yii::t('yii', '¿Estás seguro de que deseas
eliminar este elemento?'),
      'data-method' => 'post',
      'class' => 'btn btn-danger btn-xs',
    ]);
  }
];
```

```
        },
      ],
    ],
  [
    'label' => 'Formatos Incidencias',
    'format' => 'raw',
    'value' => function ($model) { //DROPDOWN PARA MOSTRAR LAS
      ACCIONES QUE SE PUEDEN REALIZAR CON LOS EMPLEADOS
      return '
        <div class="btn-group dropleft">
          <button class="btn btn-success dropdown-toggle"
            type="button" id="dropdownMenuButton" data-toggle="dropdown"
            aria-haspopup="true" aria-expanded="false">
            Crear
          </button>
          <div class="dropdown-menu " aria-labelledby="dropdownMenuButton">
            <div class="dropdown-submenu">
              <a class="dropdown-item dropdown-toggle"
                href="#">CITA MEDICA</a>
              <ul class="dropdown-menu ">
                <li> . Html::a('Ver', Url::to(['cita-medica/historial',
                  'empleado_id' => $model->id]), ['class' => 'dropdown-item text-primary',
                  'data-pjax' => '0']) . '</li>
                <div class="dropdown-divider"></div>

                <li> . Html::a('Crear', Url::to(['cita-medica/create',
                  'empleado_id' => $model->id]), ['class' => 'dropdown-item text-primary',
                  'data-pjax' => '0']) . '</li>

              </ul>
            </div>
            <div class="dropdown-submenu">
              <a class="dropdown-item dropdown-toggle"
                href="#">PERMISO FUERA DEL TRABAJO</a>
              <ul class="dropdown-menu ">
                <li> . Html::a('Ver',
                  Url::to(['permiso-fuera-trabajo/historial', 'empleado_id' => $model->id]), ['class' => 'dropdown-item text-primary', 'data-pjax' => '0']) . '</li>
                <div class="dropdown-divider"></div>

                <li> . Html::a('Crear',
                  Url::to(['permiso-fuera-trabajo/create', 'empleado_id' => $model->id]), ['class' => 'dropdown-item text-primary', 'data-pjax' => '0']) . '</li>
```

```
</ul>
</div>
<div class="dropdown-submenu">
    <a class="dropdown-item dropdown-toggle"
        href="#">COMISIÓN ESPECIAL</a>
    <ul class="dropdown-menu">
        <li>' . Html::a('Ver', Url::to(['comision-especial/historial',
            'empleado_id' => $model->id]), ['class' => 'dropdown-item text-primary',
            'data-pjax' => '0']) . '</li>
        <div class="dropdown-divider"></div>

        <li>' . Html::a('Crear',
            Url::to(['comision-especial/create', 'empleado_id' => $model->id]), ['class' =>
            'dropdown-item text-primary', 'data-pjax' => '0']) . '</li>
        </ul>
    </div>
    <div class="dropdown-submenu">
        <a class="dropdown-item dropdown-toggle"
            href="#">CAMBIO DE DÍA LABORAL</a>
        <ul class="dropdown-menu">
            <li>' . Html::a('Ver', Url::to(['cambio-dia-laboral/historial',
                'empleado_id' => $model->id]), ['class' => 'dropdown-item text-primary',
                'data-pjax' => '0']) . '</li>
            <div class="dropdown-divider"></div>

            <li>' . Html::a('Crear',
                Url::to(['cambio-dia-laboral/create', 'empleado_id' => $model->id]), ['class' =>
                'dropdown-item text-primary', 'data-pjax' => '0']) . '</li>
            </ul>
        </div>
        <div class="dropdown-submenu">
            <a class="dropdown-item dropdown-toggle"
                href="#">CAMBIO DE HORARIO DE TRABAJO</a>
            <ul class="dropdown-menu">
                <li>' . Html::a('Ver',
                    Url::to(['cambio-horario-trabajo/historial', 'empleado_id' => $model->id]),
                    ['class' => 'dropdown-item text-primary', 'data-pjax' => '0']) . '</li>
                <div class="dropdown-divider"></div>

                <li>' . Html::a('Crear',
                    Url::to(['cambio-horario-trabajo/create', 'empleado_id' => $model->id]),
                    ['class' => 'dropdown-item text-primary', 'data-pjax' => '0']) . '</li>
                </ul>
            </div>
            <div class="dropdown-submenu">
```

```
<a class="dropdown-item dropdown-toggle"
  href="#">PERMISO ECONÓMICO</a>
  <ul class="dropdown-menu">
    <li>' . Html::a('Ver',
      Url::to(['permiso-economico/historial', 'empleado_id' => $model->id]), ['class'
      => 'dropdown-item text-primary', 'data-pjax' => '0']) . '</li>
      <div class="dropdown-divider"></div>

      <li>' . Html::a('Crear',
        Url::to(['permiso-economico/create', 'empleado_id' => $model->id]), ['class' =>
        'dropdown-item text-primary', 'data-pjax' => '0']) . '</li>
      </ul>
    </div>
    <div class="dropdown-submenu">
      <a class="dropdown-item dropdown-toggle"
        href="#">PERMISO SIN GOCE DE SUELDO</a>
        <ul class="dropdown-menu">
          <li>' . Html::a('Ver',
            Url::to(['permiso-sin-sueldo/historial', 'empleado_id' => $model->id]), ['class'
            => 'dropdown-item text-primary', 'data-pjax' => '0']) . '</li>
          <div class="dropdown-divider"></div>

          <li>' . Html::a('Crear',
            Url::to(['permiso-sin-sueldo/create', 'empleado_id' => $model->id]), ['class' =>
            'dropdown-item text-primary', 'data-pjax' => '0']) . '</li>
          </ul>
        </div>
        <div class="dropdown-submenu">
          <a class="dropdown-item dropdown-toggle"
            href="#">CAMBIO DE PERIODO VACACIONAL</a>
            <ul class="dropdown-menu">
              <li>' . Html::a('Ver',
                Url::to(['cambio-periodo-vacacional/historial', 'empleado_id' => $model->id]),
                ['class' => 'dropdown-item text-primary', 'data-pjax' => '0']) . '</li>
              <div class="dropdown-divider"></div>

              <li>' . Html::a('Crear',
                Url::to(['cambio-periodo-vacacional/create', 'empleado_id' => $model->id]),
                ['class' => 'dropdown-item text-primary', 'data-pjax' => '0']) . '</li>
              </ul>
            </div>
            <div class="dropdown-submenu">
              <a class="dropdown-item dropdown-toggle"
                href="#">REPORTE DE TIEMPO EXTRA</a>
                <ul class="dropdown-menu">
```

```
<li>' . Html::a('Ver',
Url::to(['reporte-tiempo-extra/historial', 'empleado_id' => $model->id]), ['class'
=> 'dropdown-item text-primary', 'data-pjax' => '0']) . '</li>
<div class="dropdown-divider"></div>

<li>' . Html::a('Crear',
Url::to(['reporte-tiempo-extra/create', 'empleado_id' => $model->id]), ['class'
=> 'dropdown-item text-primary', 'data-pjax' => '0']) . '</li>
</ul>
</div>
</div>
</div>';
},
'contentOptions' => ['class' => 'small-font'],
],
],
'summaryOptions' => ['class' => 'summary mb-2'],
'pager' => [
'class' => 'yii\\bootstrap4\\LinkPager',
],
]);
?>

<?php } ?>

</div>
<script>
$(document).ready(function(){
$('.dropdown-submenu').on('mouseenter', function() {
$(this).children('.dropdown-menu').show();
}).on('mouseleave', function() {
$(this).children('.dropdown-menu').hide();
});
//SCRIPT QUE RESPONDE A LAS ACCIONES DEL DROPPDOWN
</script>
<?php Pjax::end(); ?>
</div>
<br>
<br><br><br> <br>
<?php $this->endBlock(); ?>
```

```
<?php $this->beginBlock('block-junta-gobierno'); ?>

<?php
//BLOQUE PARA CARGAR UNA LISTA QUE MUESTRE
LOS EMPLEADOS
//QUE SE ENCUENTRAN COMO DIRECTORES, JEFES
DE UNIDA O DEPARTAMENTO
$this->beginBlock('block-view-junta-gobierno'); ?>

<div class="card">

    <div class="card-header bg-info text-white">
        <h3>Lista de directores, jefes de unidad y jefes de
departamento</h3>
    </div>
    <div class="card-body">
        <?php
            $searchModel = new JuntaGobiernoSearch();
            $params = Yii::$app->request->queryParams;
            $params[$searchModel->formName()]['id'] =
$JuntaGobiernoModel->id;
            $dataProvider = $searchModel->search($params);
        ?>

        <?php Pjax::begin(); ?>

        <?= GridView::widget([
            'dataProvider' => $dataProvider,
            'filterModel' => $searchModel,
            'columns' => [
                'class' => 'yii\grid\SerialColumn',
                // 'id',
                [
                    'attribute' => 'foto',
                    'format' => 'html',
                    'filter' => false,
                    'value' => function ($JuntaGobiernoModel) {
                        if ($JuntaGobiernoModel->empleado->foto) {
                            $urlImagen =
Yii::$app->urlManager->createUrl(['empleado/foto-empleado', 'id' =>
$JuntaGobiernoModel->empleado->id]);
                            return Html::img($urlImagen, ['width' =>
'80px', 'height' => '80px']);
                        }
                    }
                ]
            ]
        ]); ?>
    </div>
</div>
```

```
        return null;
    },
],
[
    'label' => 'Empleado',
    'attribute' => 'empleado_id',
    'value' => function ($juntaGobiernoModel) {
        return
    $juntaGobiernoModel->empleado->nOMBRE . ' ' .
    $juntaGobiernoModel->empleado->apellido;

    },
],
[
    'label' => 'Dirección',
    'attribute' => 'cat_direccion_id',
    'value' => function ($juntaGobiernoModel) {
        return
    $juntaGobiernoModel->catDireccion->nOMBRE_direccion;
    },
],
'nivel_jerarquico',

[
    'label' => 'Nombramiento',
    'attribute' => 'empleado_id',
    'value' => function ($juntaGobiernoModel) {
        return
    $juntaGobiernoModel->empleado->informacionLaboral->catPuesto->nOMBRE_puesto;
    },
],
[
    'class' =>
'hail812\adminlte3\yii\grid\ActionColumn',
    'template' => '{view} ',
    'buttons' => [
        'view' => function ($url,
    $juntaGobiernoModel) {
            return Html::a('<i class="far fa-eye"></i>',
['empleado/view', 'id' => $juntaGobiernoModel->empleado_id], [
                '/target' => '_blank',
                'title' => 'Ver empleado',
            ]
        }
    ]
]
```

```
'class' => 'btn btn-info btn-xs',
'data-pjax' => "0"
```

```
]);
},
```

```
],
],
],
],
'summaryOptions' => ['class' => 'summary mb-2'],
'pager' => [
    'class' => 'yii\bootstrap4\LinkPager',
],
]); ?>
```

```
<?php Pjax::end(); ?>
```

```
</div>
```

```
</div>
```

```
<?php $this->endBlock(); ?>
<?php echo TabsX::widget([
    'enableStickyTabs' => true,
    'options' => ['class' => 'custom-tabs-2'],
    'items' => [
```

```
],
'position' => TabsX::POS_ABOVE,
'align' => TabsX::ALIGN_CENTER,
// 'bordered'=>true,
'encodeLabels' => false
```

```
]);
```

```
?>
```

```
<?php $this->endBlock(); ?>
```

```
</div>
<!--.col-md-12-->

<?php
$tabs = [
[
    'label' => 'Empleados',
    'content' => $this->blocks['block-empleados'],
    'active' => true,
    'options' => [
        'id' => 'empleados',
    ],
],
];
//PERMISO
if (Yii::$app->user->can('ver-lista-directores-jefes')) {
    $tabs[] = [
        'label' => 'Directores y Jefes',
        'content' => $this->blocks['block-view-junta-gobierno'],
        'options' => [
            'id' => 'junta_gobierno',
        ],
    ];
}
//CARGAR TABS
echo TabsX::widget([
    'enableStickyTabs' => true,
    'options' => ['class' => 'custom-tabs-2'],
    'items' => $tabs,
    'position' => TabsX::POS_ABOVE,
    'align' => TabsX::ALIGN_CENTER,
    'encodeLabels' => false,
]);
?>

</div>

</div>

</div>
```

- view.

Descripción. Código que contiene las partes principales del funcionamiento del sistema. En este código se renderizan diferentes vistas que cargan formularios de distintos modelos, para mostrar los registros relacionados acerca de la información completa de los empleados. (Información personal, laboral, de contacto, vacacional, expediente médico, documentos personales y médicos, historial de solicitudes). También se incluyen distintos permisos para poder realizar ciertas acciones o mostrar información.

```
<?php
//IMPORTACIONES
use app\models\Solicitud;
use yii\bootstrap5\Alert;
use yii\helpers\Html;
use kartik\file\TextInput;
use yii\web\View;
use kartik\form\ActiveForm;
use kartik\tabs\TabsX;

/* @var $this yii\web\View */
/* @var $model app\models\Empleado */

//$activeTab = Yii::$app->request->get('tab', 'info_p');
//CARGAR ESTILOS
$this->registerCssFile('@web/css/site.css', ['position' => View::POS_HEAD]);
$this->registerCssFile('@web/css/grid-view.css', ['position' =>
View::POS_HEAD]);

$this->registerJsFile('@web/js/municipios.js', ['depends' =>
[yii\web\JqueryAsset::class]]);
$this->registerJsFile('@web/js/select_estados.js', ['depends' =>
[yii\web\JqueryAsset::class]]);
if (Yii::$app->user->can('medico') || Yii::$app->user->can('gestor-rh')) :

    $this->title = 'Empleado: ' . $model->nombre . ' ' . $model->apellido;
    $this->params['breadcrumbs'][] = ['label' => 'Empleados', 'url' => ['index']];
endif;

yii\web\YiiAsset::register($this);
$currentTime = date('Y-m-d');
$antecedentesExistentes = [];
$observacionGeneral = "";
$descripcionAntecedentes = "";
```

```
$modelAntecedenteNoPatologico = new
\app\models\AntecedenteNoPatologico();
$modelExploracionFisica = new \app\models\ExploracionFisica();
$editable = Yii::$app->user->can('editar-expediente-medico');

//CARGAR ANTECEDENTES
if ($antecedentes) {
    foreach ($antecedentes as $antecedente) {

        $antecedentesExistentes[$antecedente->cat_antecedente_hereditario_id][$a
ntecedente->parentezco] = true;
        if (empty($observacionGeneral)) {
            $observacionGeneral = $antecedente->observacion;
        }
    }
}

// Si ya existe un antecedente patológico se obtiene el registro
$modelAntecedentePatologico =
\app\models\AntecedentePatologico::findOne(['expediente_medico_id' =>
$expedienteMedico->id]);
if (!$modelAntecedentePatologico) {
    $modelAntecedentePatologico = new
\app\models\AntecedentePatologico();
    $modelAntecedentePatologico->expediente_medico_id =
$expedienteMedico->id;
}

// Obtener antecedentes no patológicos
$modelAntecedenteNoPatologico =
\app\models\AntecedenteNoPatologico::findOne(['expediente_medico_id' =>
$expedienteMedico->id]);
if (!$modelAntecedenteNoPatologico) {
    $modelAntecedenteNoPatologico = new
\app\models\AntecedenteNoPatologico();
    $modelAntecedenteNoPatologico->expediente_medico_id =
$expedienteMedico->id;
}

$modelExploracionFisica =
\app\models\ExploracionFisica::findOne(['expediente_medico_id' =>
$expedienteMedico->id]);
if (!$modelExploracionFisica) {
    $modelExploracionFisica = new \app\models\ExploracionFisica();
```

```
$modelExploracionFisica->expediente_medico_id =
$expedienteMedico->id;
}

$modelInterrogatorioMedico =
\app\models\InterrogatorioMedico::findOne(['expediente_medico_id' =>
$expedienteMedico->id]);
if (!$modelInterrogatorioMedico) {
    $modelInterrogatorioMedico = new \app\models\InterrogatorioMedico();
    $modelInterrogatorioMedico->expediente_medico_id =
$expedienteMedico->id;
}

$modelAntecedenteObstrectico =
\app\models\AntecedenteObstrectico::findOne(['expediente_medico_id' =>
$expedienteMedico->id]);
if (!$modelAntecedenteObstrectico) {
    $modelAntecedenteObstrectico = new
\app\models\AntecedenteObstrectico();
    $modelAntecedenteObstrectico->expediente_medico_id =
$expedienteMedico->id;
}

$modelAlergia = \app\models\Alergia::findOne(['expediente_medico_id' =>
$expedienteMedico->id]);
if (!$modelAlergia) {
    $modelAlergia = new \app\models\Alergia();
    $modelAlergia->expediente_medico_id = $expedienteMedico->id;
}

?>
<link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/sweetalert2@11/dist/sweetalert2.min.css">
<script
src="https://cdn.jsdelivr.net/npm/sweetalert2@11/dist/sweetalert2.all.min.js"><
/script>

<script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
<script>
    function showAlert(title, text) {
        Swal.fire({
            icon: 'warning',
            title: title,
```

```
text: text,
});
}
</script>

<div class="container-fluid">
    <div class="row justify-content-center">
        <div class="col-md-12">
            <div class="card bg-light">
                <div class="card-header gradient-info text-white">
                    <div class="d-flex align-items-center position-relative ml-4">
                        <div class="bg-light p-1 rounded-circle custom-shadow mt-4"
style="width: 140px; height: 140px; position: relative;">
                            <?= Html::img(['empleado/foto-empleado', 'id' =>
$model->id], [
                                'class' => 'rounded-circle',
                                'style' => 'width: 130px; height: 130px;'
                            ]) ?>
                            <?php
//PERMISO PARA PODER CAMBIAR LA FOTO DEL
EMPLEADO
if (Yii::$app->user->can('modificar-informacion-empleados'))
: ?>
                            <?= Html::button('<i class="fas fa-edit"></i>', [
                                'class' => 'btn btn-dark position-absolute',
                                'style' => 'top: 5px; right: 5px; padding: 5px 10px;',
                                'data-toggle' => 'modal',
                                'data-target' => '#changePhotoModal'
                            ]) ?>
                            <?php endif; ?>
                        </div>
                    <div class="ml-4">
                        <div class="alert alert-light mb-0" role="alert">
                            <h3 class="mb-0"><?= $model->nombre ?> <?= $model->apellido
?></h3>
                            <h5 class="mb-0">Número de empleado: <?=
$model->numero_empleado ?></h5>
                        </div>
                    </div>
                </div>
            </div>
        </div>
    </div>
</div>
<?php
```

```
//PERMISO PARA MOSTRAR LOS BOTONES DE CREAR  
CONSULTA Y CITA MEDICA  
if (Yii::$app->user->can('crear-consulta-medica')) : ?>  
  
<?php if ($model->expedienteMedico) : ?>  
    <?= Html::a('Nueva consulta <i class="fa fa-user-md" ></i>',  
['consulta-medica/create', 'expediente_medico_id' =>  
$model->expedienteMedico->id], [  
    'class' => 'btn btn-dark float-right fa-lg'  
]) ?>  
  
<?php endif; ?>  
  
  
<?php if ($model->id) : ?>  
  
    <?= Html::a('Nueva cita medica <i class="fa fa-plus-square" ></i>', ['cita-medica/create', 'empleado_id' => $model->id], [  
        'class' => 'btn btn-light mr-3 float-right fa-lg'  
]) ?>  
  
<?php endif; ?>  
  
<?php endif; ?>  
  
</div>  
  
  
<div class="modal fade" id="changePhotoModal" tabindex="-1"  
role="dialog" aria-labelledby="changePhotoModalLabel" aria-hidden="true">  
    <div class="modal-dialog modal-md" role="document">  
        <div class="modal-content">  
            <div class="modal-header">  
                <h5 class="modal-title"  
id="changePhotoModalLabel">Cambiar Foto de Perfil</h5>  
  
                <button type="button" class="close"  
data-dismiss="modal" aria-label="Close">  
                    <span aria-hidden="true">&times;</span>  
                </button>
```

```
</div>
<div class="modal-body">
<?php
//FORMULARIO PARA SUBIR Y HACER EL CAMBIO DE
FOTO DEL PERFIL
$form = ActiveForm::begin([
    'action' => ['empleado/change-photo', 'id' =>
$model->id],
    'options' => ['enctype' => 'multipart/form-data']
]); ?>

<?= $form->field($model,
'foto')->widget(FileInput::classname(), [
    'options' => ['accept' => 'image/*'],
    'pluginOptions' => [
        'showPreview' => true,
        'showCaption' => true,
        'showRemove' => true,
        'showUpload' => false,
        'showCancel' => false,
        'browseClass' => 'btn btn-primary btn-block',
        'browseLabel' => 'Seleccionar Foto'
    ]
]); ?>

<div class="form-group">
<?= Html::submitButton('Subir', ['class' => 'btn
btn-primary']) ?>
</div>

<?php ActiveForm::end(); ?>
</div>
</div>
</div>
</div>

<div class="card-body">
<div class="row">
<div class="col-md-12">
<div class="d-flex align-items-center mb-3 float-right">
```

```
<?php

//ALERTA
foreach (Yii::$app->session->getAllFlashes() as $type =>
$message) {
    if ($type === 'error') {
        echo Alert::widget([
            'options' => ['class' => 'alert-danger'],
            'body' => $message,
        ]);
    } else {
        echo Alert::widget([
            'options' => ['class' => 'alert-' . $type],
            'body' => $message,
        ]);
    }
}
?>
<?php
//VERIFICA EL ESTATUS DE PRIMERA_REVISION
if ($model->expedienteMedico !== null) {
    if ($model->expedienteMedico->primera_revision === 0) {
        echo '

            <div class="alert alert-warning mb-0" role="alert">
                <h6> <i class="fas fa-exclamation-triangle" style="color: #dd0000 "></i> PENDIENTE DE PRIMERA REVISION MEDICA</h6> </div>';
    }
}
?>
</div>
<?php $this->beginBlock('datos'); ?>
<?php $this->beginBlock('info_p'); ?>
<?php
// Otros contenidos de la vista principal

// Incluir el contenido del nuevo archivo de vista
echo $this->render('informacion_personal', [
    'model' => $model,
]);

?>
<?php $this->endBlock(); ?>
```

```
<?php $this->beginBlock('info_c'); ?>

<?php $this->endBlock(); ?>
<!-- INFOLABORAL-->
<?php $this->beginBlock('info_l'); ?>
<?php
// Otros contenidos de la vista principal

// Incluir el contenido del nuevo archivo de vista
echo $this->render('informacion_laboral', [
    'model' => $model,

]);


?>

<?php $this->endBlock(); ?>

<?php $this->beginBlock('info_vacacional'); ?>
<?php
// Otros contenidos de la vista principal

// Incluir el contenido del nuevo archivo de vista
echo $this->render('informacion_vacacional', [
    'model' => $model,
    'currentDate' => $currentDate,
    'historial' => $historial,
]);


?>

<?php $this->endBlock(); ?>

<?php $this->beginBlock('info_solicitudes'); ?>
<?php
// Otros contenidos de la vista principal

// Incluir el contenido del nuevo archivo de vista
echo $this->render('informacion_solicitudes', [
    'model' => $model,
    'searchModel' => $searchModel,
    'dataProvider' => $dataProvider,
    'searchModelConsultas' => $searchModelConsultas,
    'dataProviderConsultas' => $dataProviderConsultas,
```

```
];
?>

<?php $this->endBlock(); ?>

<?php
// Función para contar las nuevas solicitudes
// Función para contar las nuevas solicitudes de un empleado
específico
function countNewSolicitudes($empleado_id)
{
    return Solicitud::find()->where(['status' => 'Nueva', 'empleado_id' =>
$empleado_id])->count();
}

// Contar las nuevas solicitudes
// Obtener el empleado_id del modelo
$empleado_id = $model->id;

// Contar las nuevas solicitudes para el empleado específico
$newSolicitudesCount = countNewSolicitudes($empleado_id);

// Configurar las pestañas
$tabs = [
[
    'label' => 'Información personal',
    'content' => $this->blocks['info_p'],
    'options' => [
        'id' => 'informacion_personal',
    ],
],
[
    'label' => 'Información laboral',
    'content' => $this->blocks['info_l'],
    'options' => [
        'id' => 'informacion_laboral',
    ],
],
];
if (Yii::$app->user->can('ver-informacion-vacacional')) {
    $tabs[] = [
        'label' => 'Vacaciones',
    ];
}
```

```
'content' => $this->blocks['info_vacacional'],
'options' => [
    'id' => 'informacion_vacaciones',
],
];
}

if (Yii::$app->user->can('medico') || Yii::$app->user->can('gestor-rh')) {
    $tabs[] = [
        'label' => 'Solicitudes' . ($newSolicitudesCount > 0 ? " <span
class='badge badge-danger'>$newSolicitudesCount</span>" : ""),
        'content' => $this->blocks['info_solicitudes'],
        'options' => [
            'id' => 'informacion_solicitudes',
        ],
    ];
} else {
    $tabs[] = [
        'label' => 'Solicitudes',
        'content' => $this->blocks['info_solicitudes'],
        'options' => [
            'id' => 'informacion_solicitudes',
        ],
    ];
}
// CARGAR TABS CON LOS BLOQUES QUE RENDERIZAN LAS VISTAS
DE INFORMACION
echo TabsX::widget([
    'enableStickyTabs' => true,
    'options' => ['class' => 'nav-tabs-custom'],
    'items' => $tabs,
    'position' => TabsX::POS_ABOVE,
    'align' => TabsX::ALIGN_CENTER,
    'encodeLabels' => false,
]);
?>
```

```
<?php $this->beginBlock('info_consultas'); ?>
<?php
// RENDERIZAR HISTORIAL DE CONSULTAS MEDICAS
echo $this->render('informacion_consultas', [
    'model' => $model,
    'searchModelConsultas' => $searchModelConsultas,
    'dataProviderConsultas' => $dataProviderConsultas,
```

```
];
?>
<?php $this->endBlock(); ?>

<?php $this->beginBlock('expediente'); ?>
<?php
//RENDERIZAR DOCUMENTOS DEL EMPLEADO
echo $this->render('documentos_empleado', [
    'model' => $model,
    'searchModel' => $searchModel,
    'dataProvider' => $dataProvider,
    'documentos' => $documentos,
    'documentoModel' => $documentoModel,

]);
?>

<?php $this->endBlock(); ?>
<?php
//BLOQUE DONDE SE RENDERIZA LA INFORMACION
MEDICA DEL
//EMPLEADO Y SUS RESPECTIVOS FORMULARIOS
    $this->beginBlock('expediente_medico');

?>

<?php $this->beginBlock('documento-medico'); ?>
<?php
// RENDERIZA DOCUMENTO MEDICOS

echo $this->render('documento_medico_empleado', [
    'model' => $model,
    'searchModel' => $searchModel,
    'dataProvider' => $dataProvider,
    'documentoMedicoModel' => $documentoMedicoModel,

]);
?>
<?php $this->endBlock(); ?>
```

```
<?php $this->beginBlock('antecedentes'); ?>

<!-- Bloque de antecedentes hereditarios -->
<?php $this->beginBlock('hereditarios'); ?>
<?php

echo $this->render('antecedente_hereditario', [
    'model' => $model,
    'catAntecedentes' => $catAntecedentes,
    'antecedentes' => $antecedentes,
    ]);

?>
<?php $this->endBlock(); ?>

<!-- Bloque de antecedentes patológicos -->
<?php $this->beginBlock('patologicos'); ?>
<?php

echo $this->render('antecedente_patologico', [
    'model' => $model,
    'antecedentePatologico' => $antecedentePatologico,
    ]);

?>
<?php $this->endBlock(); ?>

<?php $this->beginBlock('no_patologicos'); ?>
<?php

echo $this->render('antecedente_no_patologico', [
    'model' => $model,
    'antecedenteNoPatologico' =>
    $antecedenteNoPatologico,
    ]);

?>
```

```
?>
<?php $this->endBlock(); ?>

<?php $this->beginBlock('perinatales'); ?>
<?php

echo $this->render('antecedente_perinatal', [
    'model' => $model,

    'AntecedentePerinatal' => $AntecedentePerinatal,
    'expedienteMedico' => $expedienteMedico,


]);;
?>
<?php $this->endBlock(); ?>

<?php $this->beginBlock('ginecologicos'); ?>
<?php

echo $this->render('antecedente_ginecologico', [
    'model' => $model,

    'AntecedenteGinecologico' =>
$AntecedenteGinecologico,


    'expedienteMedico' => $expedienteMedico,


]);;
?>
<?php $this->endBlock(); ?>

<?php $this->beginBlock('obstrecticos'); ?>
<?php

echo $this->render('antecedente_obstrectico', [
    'model' => $model,
```

'AntecedenteObstrectico' => \$AntecedenteObstrectico,

```
]);
?>
<?php $this->endBlock(); ?>

<!-- TABS EXPEDIENTE MEDICO -->

<?php echo TabsX::widget([
    'enableStickyTabs' => true,
    'options' => ['class' => 'nav-tabs-custom'],
    'items' => [
        [
            'label' => 'Hereditarios',
            'content' => $this->blocks['hereditarios'],
            //active => true,
            'options' => [
                'id' => 'hereditarios',
            ],
        ],
        [
            'label' => 'Patologicos',
            'content' => $this->blocks['patologicos'],
            'options' => [
                'id' => 'patologicos',
            ],
        ],
        [
            'label' => 'No Patologicos',
            'content' => $this->blocks['no_patologicos'],
            'options' => [
                'id' => 'nopatologicos',
            ],
        ],
    ],
]);
```

```
],
[
    'label' => 'Perinatales',
    'content' => $this->blocks['perinatales'],

    'options' => [
        'id' => 'perinatales',
    ],
],

],
[
    'label' => 'Ginecologicos',
    'content' => $this->blocks['ginecologicos'],

    'options' => [
        'id' => 'ginecologicos',
    ],
],

],
[
    'label' => 'Obstetricicos',
    'content' => $this->blocks['obstetricicos'],

    'options' => [
        'id' => 'obstetricicos',
    ],
],

],
],
'position' => TabsX::POS_ABOVE,
'align' => TabsX::ALIGN_CENTER,
//  'bordered'=>true,
'encodeLabels' => false
]);
```

?>

```
<?php $this->endBlock(); ?>
```

```
<?php $this->beginBlock('alergias'); ?>
<?php
```

```
echo $this->render('alergias', [
    'model' => $model,
```

```
    'Alergia' => $Alergia,
```

```
]);
```

```
?>
```

```
<?php $this->endBlock(); ?>
```

```
<?php $this->beginBlock('exploracion_fisica'); ?>
<?php
```

```
echo $this->render('exploracion_fisica', [
    'model' => $model,
```

```
    'ExploracionFisica' => $ExploracionFisica,
```

```
]);
```

```
?>
```

```
<?php $this->endBlock(); ?>
```

```
<?php $this->beginBlock('interrogatorio_medico'); ?>
<?php
```

```
echo $this->render('interrogatorio_medico', [
    'model' => $model,
```

```
    'InterrogatorioMedico' => $InterrogatorioMedico,
```

```
]);
?>
<?php $this->endBlock(); ?>

<!-- TABS PRINCIPAL EXPEDIENTE MEDICO-->
<?php

// Construir las pestañas
$itemss = [
    [
        'label' => 'Antecedentes',
        'content' => $this->blocks['antecedentes'],
        'options' => ['id' => 'antecedentes'],
    ],
    [
        'label' => 'Exploración Física',
        'content' => $this->blocks['exploracion_fisica'],
        'options' => ['id' => 'exploracion_fisica'],
    ],
    [
        'label' => 'Interrogatorio Médico',
        'content' => $this->blocks['interrogatorio_medico'],
        'options' => ['id' => 'interrogatorio_medico'],
    ],
    [
        'label' => 'Alergias',
        'content' => $this->blocks['alergias'],
        'options' => ['id' => 'alergias'],
    ],
];
;

if (Yii::$app->user->can('ver-documentos-medicos')) {
    $itemss[] = [
        'label' => 'Documentos Médicos',
        'content' => $this->blocks['documento-medico'],
        'options' => ['id' => 'documentos-medicos'],
    ];
}

// Mostrar el widget de TabsX
```

```
echo TabsX::widget([
    'enableStickyTabs' => true,
    'options' => ['class' => 'nav-tabs-custom'],
    'items' => $items,
    'position' => TabsX::POS_ABOVE,
    'align' => TabsX::ALIGN_LEFT,
    'encodeLabels' => false,
]);
?>

<?php $this->endBlock(); ?>

<!-- antecedentes-->

<?php $this->endBlock(); ?>

</div>
<!--.col-md-12-->

<!-- TABS PRINCIPAL -->
<?php
$tabs = [
[
    'label' => 'Información',
    'content' => $this->blocks['datos'],
    'active' => true,
    'options' => [
        'id' => 'datos',
    ],
],
];
if (Yii::$app->user->can('ver-expediente-medico')) {
    $tabs[] = [
        'label' => 'Expediente Medico',
        'content' => $this->blocks['expediente_medico'],
        'options' => [
            'id' => 'expediente_medico',
        ],
    ];
}
```

```
        ],
    ];
}
if (Yii::$app->user->can('ver-documentos')) {
    $tabs[] = [
        'label' => 'Documentos',
        'content' => $this->blocks['expediente'],
        'options' => [
            'id' => 'documentos',
        ],
    ];
}
if (Yii::$app->user->can('ver-consultas-medicas')) {
    $tabs[] = [
        'label' => 'Consultas medicas',
        'content' => $this->blocks['info_consultas'],
        'options' => [
            'id' => 'informacion_consultas',
        ],
    ];
}

echo TabsX::widget([
    'enableStickyTabs' => true,
    'options' => ['class' => 'nav-tabs-custom'],
    'items' => $tabs,
    'position' => TabsX::POS_ABOVE,
    'align' => TabsX::ALIGN_LEFT,
    'encodeLabels' => false,
]);
?>

</div>
<!--.row-->

</div>
<!--.card-body-->

</div>
<!--.card-->
```

```
</div>
<!--.col-md-10-->

</div>
<!--.row-->
<script>
    //SCRIPT PARA SEGURARSE DE QUE SE RECARGUEN BIEN
    //LAS PESTAÑAS EN BASE A SU ID
    $(document).ready(function() {
        function activateTabFromHash() {
            var hash = window.location.hash;

            if (hash) {
                var tabId = hash.replace('#', "");

                var tabLink = $('a[href="#" + hash + "]');

                if (tabLink.length) {
                    tabLink.tab('show');

                    tabLink.parents('.tab-pane').each(function() {
                        var parentTabId = $(this).attr('id');
                        $('a[href="#" + parentTabId + "]').tab('show');
                    });
                }
            }
        }

        activateTabFromHash();

        $(window).on('hashchange', function() {
            activateTabFromHash();
        });

        $('a[data-toggle="tab"]').on('shown.bs.tab', function(e) {
            window.location.hash = $(e.target).attr('href');
        });
    });
</script>
</div>
```

- informacion\_personal.

Descripción. Código que tiene la funcionalidad de mostrar la información personal y de contacto del empleado. Se encuentran

permisos que permiten modificar o añadir la información mediante formularios.

```
<?php
//IMPORTACIONES
use app\models\CatNivelEstudio;

use yii\helpers\Html;

use yii\widgets\Pjax;

use kartik\form\ActiveForm;

use yii\helpers\ArrayHelper;

use kartik\select2\Select2;

?

<br>

<?php Pjax::begin([
    'id' => 'pjax-update-info',
    'options' => ['pushState' => false],
]); ?>

<?php $form = ActiveForm::begin([
    'action' => ['actualizar-informacion', 'id' => $model->id],
    'options' => ['id' => 'personal-info-form']
]); ?>

<div id="loading-spinner" style="color: #000000;">
    <i class="fa fa-spinner fa-spin fa-2x" style="color: #000000;"></i>
    Procesando...
</div>
<?php
//PERMISO PARA HABILITAR LA EDICIÓN DE LA INFORMACIÓN
if (Yii::$app->user->can('modificar-informacion-empleados')) : ?>
    <div id="floating-buttons">
        <button type="button" id="edit-button-personal" class="btn
btn-warning"><i class="fa fa-edit"></i></button>
        <?= Html::submitButton('<i class="fa fa-save"></i>', ['class' => 'btn
btn-success', 'id' => 'save-button-personal', 'style' => 'display:none;']) ?>
```

```
<button type="button" id="cancel-button-personal" class="btn  
btn-danger" style="display:none;"><i class="fa fa-times"></i></button>
```

```
</div>
<?php endif; //INFORMACION PERSONAL ?>
```

```
<div class="card bg-light">
    <div class="card-header bg-info text-white">
        <h6>Personal</h6>

    </div>
    <div class="card-body">
        <div class="row">
            <div class="col-6 col-sm-2">
                <?= $form->field($model,
'numero_empleado')->input('number', [
                    'readonly' => true,
                    'maxlength' => 4,
                    'class' => 'form-control',
                    'autocomplete'=>"off",
                    'title' => $model->numero_empleado
                )->label('No. Empleado:', [
                    'title' => $model->numero_empleado
                ]) ?>
            </div>

            <div class="col-6 col-sm-4">
                <?= $form->field($model, 'nombre')->tex
                    'readonly' => true,
                    'class' => 'form-control',
                    'title' => $model->nombre,
                    'autocomplete'=>"off"
                )->label('Nombre', [
                    'title' => $model->nombre
                ]) ?>

                <?= $form->field($model, 'apellido')->te
                    'readonly' => true,
                    'class' => 'form-control',
                    'title' => $model->apellido,
                    'autocomplete'=>"off"
                )->label('Apellido', [
                    'title' => $model->apellido
                ]) ?>
```

```
        ]) ?>
    </div>

    <?php //echo
Html::label($model->expedienteMedico->antecedenteNoPatologico,
'religion');

?>
<div class="col-6 col-sm-2">
    <?= $form->field($model,
'fecha_nacimiento)->input('date', [
        'disabled' => true,
        'title' => $model->fecha_nacimiento
])->label('Fecha Nacimiento', [
        'title' => $model->fecha_nacimiento
]) ?>

    <?= $form->field($model, 'edad')->hiddenInput([
        'title' => $model->edad
])->label(false); ?>

    <div class="form-group">
        <label class="control-label" title="<?=
Html::encode($model->edad); ?>">Edad:</label>
        <p id="edad-label" title="<?=
Html::encode($model->edad); ?><?= Html::encode($model->edad); ?></p>
        </div>
    </div>

    <div class="col-6 col-sm-3">
        <?= $form->field($model, 'sexo')->dropDownList([
            'Masculino' => 'Masculino',
            'Femenino' => 'Femenino',
        ], [
            'prompt' => 'Seleccionar sexo',
            'disabled' => true,
            'class' => 'form-control',
            'title' => $model->sexo
])->label('Sexo', [
            'title' => $model->sexo
]) ?>

        <?= $form->field($model,
'estado_civil')->dropDownList([
            'Soltero/a' => 'Soltero/a',
            'Casado/a' => 'Casado/a',
        ],
        [
            'prompt' => 'Seleccionar estado civil',
            'disabled' => true,
            'class' => 'form-control',
            'title' => $model->estado_civil
])->label('Estado Civil', [
            'title' => $model->estado_civil
]) ?>
    </div>
</div>
```

```
'Separado/a' => 'Separado/a',
'Viudo/a' => 'Viudo/a',
], [
    'prompt' => 'Seleccionar estado civil',
    'disabled' => true,
    'class' => 'form-control',
    'title' => $model->estado_civil
])->label('Estado civil', [
    'title' => $model->estado_civil
]) ?>
</div>

<div class="w-100"></div>
<div class="col-6 col-sm-3">
    <?= $form->field($model, 'curp')->textInput([
        'readonly' => true,
        'maxlength' => 18,
        'class' => 'form-control',
        'title' => $model->curp,
        'autocomplete'=>"off"
    ])->label('CURP:', [
        'title' => $model->curp
    ]) ?>
</div>

<div class="col-6 col-sm-3">
    <?= $form->field($model, 'nss')->input('number', [
        'maxlength' => 11,
        'class' => 'form-control',
        'id' => 'nss-input',
        'readonly' => true,
        'title' => $model->nss,
        'autocomplete'=>"off"
    ])->label('NSS:', [
        'title' => $model->nss
    ]) ?>
</div>

<div class="col-6 col-sm-3">
    <?= $form->field($model, 'rfc')->textInput([
        'readonly' => true,
        'maxlength' => 13,
        'class' => 'form-control',
        'title' => $model->rfc,
        'autocomplete'=>"off"
    ])->label('RFC:', [
```

```
'title' => $model->rfc
]) ?>
</div>

<div class="w-100"></div>
<div class="col-6 col-sm-4">
<?= $form->field($model,
'cat_nivel_estudio_id')->widget(Select2::classname(), [
'data' => ArrayHelper::map(CatNivelEstudio::find()->all(), 'id',
'nivel_estudio'),
'options' => [
'placeholder' => 'Seleccionar Nivel de Estudio',
'disabled' => true,
'title' => $model->cat_nivel_estudio_id ?

CatNivelEstudio::findOne($model->cat_nivel_estudio_id)->nivel_estudio : "
],
'pluginOptions' => [
'allowClear' => true
],
'theme' => Select2::THEME_BOOTSTRAP,
'pluginEvents' => [
'select2:opening' => "function() {
$('.select2-selection__clear').html('<span class=\"fas fa-times\"></span>'); }",
'select2:opening' => "function() {
$('.select2-selection__clear').css('margin-left', '2px'); }",
],
])->label('Nivel de estudios',
'title' => $model->cat_nivel_estudio_id ?

CatNivelEstudio::findOne($model->cat_nivel_estudio_id)->nivel_estudio : "
]) ?>
</div>

<div class="col-6 col-sm-4">
<?= $form->field($model, 'institucion_educativa')->textInput([
'readonly' => true,
'class' => 'form-control',
'title' => $model->institucion_educativa,
'autocomplete'=>"off"
])->label('Institución Educativa', [
'title' => $model->institucion_educativa
]) ?>
</div>

<div class="col-6 col-sm-2">
```

```
<?= $form->field($model, 'profesion')->widget(Select2::classname(), [
    'data' => [
        'No tiene' => 'No tiene',
        'ING.' => 'ING.',
        'LIC.' => 'LIC.',
        'PROF.' => 'PROF.',
        'ARQ.' => 'ARQ.',
        'C.' => 'C.',
        'DR.' => 'DR.',
        'DRA.' => 'DRA.',
        'TEC.' => 'TEC.',
    ],
    'options' => [
        'prompt' => 'Seleccionar Profesión',
        'disabled' => true,
        'title' => $model->profesion
    ],
    'pluginOptions' => [
        'allowClear' => true
    ],
    'theme' => Select2::THEME_BOOTSTRAP,
])->label('Profesión', [
    'title' => $model->profesion
]) ?>
</div>

</div>
</div>

</div>

<div class="card bg-light">
    <div class="card-header bg-success text-white">
        <h6>Contacto</h6>
    </div>
    <div class="card-body">
        <div class="row">
            <div class="col-6 col-sm-3">
                <?=
                //INFORMACIÓN DE CONTACTO
                $form->field($model, 'email')->textInput([
                    'maxlength' => true,
                    'readonly' => true,
                ])
            </div>
        </div>
    </div>
</div>
```

```
'class' => 'form-control',
'title' => $model->email,
'autocomplete'=>"off"
])->label('Email', [
    'title' => $model->email
]) ?>
</div>

<div class="col-6 col-sm-2">
<?= $form->field($model, 'telefono')->textInput([
    'maxlength' => 15,
    'readonly' => true,
    'class' => 'form-control',
    'title' => $model->telefono,
    'autocomplete'=>"off"

])->label('Teléfono', [
    'title' => $model->telefono
]) ?>
</div>

<div class="col-6 col-sm-3">
<?= $form->field($model, 'colonia')->textInput([
    'maxlength' => true,
    'readonly' => true,
    'class' => 'form-control',
    'title' => $model->colonia,
    'autocomplete'=>"off"
])->label('Colonia', [
    'title' => $model->colonia
]) ?>
</div>

<div class="col-6 col-sm-3">
<?= $form->field($model, 'calle')->textInput([
    'maxlength' => true,
    'readonly' => true,
    'class' => 'form-control',
    'title' => $model->calle,
    'autocomplete'=>"off"
])->label('Calle', [
    'title' => $model->calle
]) ?>
</div>

<div class="w-100"></div>
```

```
<div class="col-6 col-sm-2">
    <?= $form->field($model, 'numero_casa')->textInput([
        'maxlength' => true,
        'readonly' => true,
        'class' => 'form-control',
        'title' => $model->numero_casa,
        'autocomplete'=>"off"
    ])->label('No. Casa:', [
        'title' => $model->numero_casa
    ]) ?>
</div>

<div class="col-6 col-sm-2">
    <?= $form->field($model, 'codigo_postal')->input('number', [
        'maxlength' => 6,
        'class' => 'form-control',
        'readonly' => true,
        'title' => $model->codigo_postal,
        'autocomplete'=>"off"
    ])->label('Código Postal:', [
        'title' => $model->codigo_postal
    ]) ?>
</div>
<div class="col-6 col-sm-4">
    <?= $form->field($model, 'estado')->widget(Select2::classname(), [
        'data' => [], // Inicialmente vacío, se llenará con JS
        'options' => [
            'placeholder' => 'Selecciona un estado',
            'id' => 'estado-dropdown',
            'disabled' => true,
            'title' => $model->estado ? $model->estado : 'No seleccionado'
        ],
        'pluginOptions' => [
            'allowClear' => true
        ],
        'theme' => Select2::THEME_BOOTSTRAP,
    ])->label('Estado:', [
        'title' => $model->estado ? $model->estado : 'No seleccionado'
    ]) ?>
</div>

<div class="col-6 col-sm-4">
    <?= $form->field($model, 'municipio')->widget(Select2::classname(), [
        'data' => [], // Inicialmente vacío
        'options' => [

```

```
'placeholder' => 'Selecciona un municipio',
'id' => 'municipio-dropdown',
'disabled' => true,
'title' => $model->municipio ? $model->municipio : 'No seleccionado'
],
'pluginOptions' => [
  'allowClear' => true
],
'theme' => Select2::THEME_BOOTSTRAP,
])->label('Municipio:', [
  'title' => $model->municipio ? $model->municipio : 'No seleccionado'
]) ?>
</div>
```

```
<h5 class="card-title">Información de Contacto de  
Emergencia</h5>
```

```
<br><br>
<div class="col-6 col-sm-4">
<?= $form->field($model, 'nombre_contacto_emergencia')->textInput([
  'maxlength' => true,
  'readonly' => true,
  'class' => 'form-control',
  'title' => $model->nombre_contacto_emergencia ?
  $model->nombre_contacto_emergencia : 'No disponible',
  'autocomplete'=>"off"
])->label('Nombre:', [
  'title' => $model->nombre_contacto_emergencia ?
  $model->nombre_contacto_emergencia : 'No disponible'
]) ?>
</div>
```

```
<div class="col-6 col-sm-4">
  <?= $form->field($model,
'relacion_contacto_emergencia')->widget(Select2::className(), [
  'data' => [
    'Padre' => 'Padre',
    'Madre' => 'Madre',
    'Esposo/a' => 'Esposo/a',
    'Hijo/a' => 'Hijo/a',
    'Hermano/a' => 'Hermano/a',
    'Compañero/a de trabajo' => 'Compañero/a de trabajo',
    'Tio/a' => 'Tio/a'
  ],
  'options' => [
    'prompt' => 'Seleccionar Parentesco',
  ]
]) ?>
</div>
```

```
'disabled' => true,
    'title' => $model->relacion_contacto_emergencia ?
$model->relacion_contacto_emergencia : 'No disponible'
],
    'pluginOptions' => [
        'allowClear' => true
],
    'theme' => Select2::THEME_BOOTSTRAP,
])->label('Relación:', [
    'title' => $model->relacion_contacto_emergencia ?
$model->relacion_contacto_emergencia : 'No disponible'
]) ?>
</div>

<div class="col-6 col-sm-3">
<?= $form->field($model, 'telefono_contacto_emergencia')->textInput([
    'maxlength' => true,
    'readonly' => true,
    'class' => 'form-control',
    'title' => $model->telefono_contacto_emergencia ?
$model->telefono_contacto_emergencia : 'No disponible',
    'autocomplete'=>"off"
])->label('Teléfono:', [
    'title' => $model->telefono_contacto_emergencia ?
$model->telefono_contacto_emergencia : 'No disponible'
]) ?>
</div>

</div>
</div>

</div>

<!-- principales-->

<?php ActiveForm::end(); ?>
<?php
$script = <<< JS
$('#personal-info-form').on('beforeSubmit', function() {
```

```
var button = $('#save-button-personal');
var spinner = $('#loading-spinner');

button.prop('disabled', true); // Deshabilita el botón
spinner.show(); // Muestra el spinner

return true; // Permite que el formulario se envíe
});

JS;
$this->registerJs($script);
?>

<script>
    // Objeto para almacenar los valores originales de los
campos
    var originalValues = {};

    // Función para almacenar los valores originales de los
campos
    function storeOriginalValues() {
        var fields =
document.querySelectorAll('#personal-info-form .form-control');
        fields.forEach(function(field) {
            originalValues[field.id] = field.value;
        });
    }

    // Función para restaurar los valores originales de los
campos
    function restoreOriginalValues() {
        var fields =
document.querySelectorAll('#personal-info-form .form-control');
        fields.forEach(function(field) {
            if (originalValues.hasOwnProperty(field.id)) {
                field.value = originalValues[field.id];
            }
        });
    }

document.getElementById('edit-button-personal').addEventListener('click',
function() {
    storeOriginalValues(); // Almacenar los valores
originales al iniciar la edición
    var fields =
document.querySelectorAll('#personal-info-form .form-control');
```

```
fields.forEach(function(field) {
    field.readOnly = false;
    field.disabled = false;
});

document.getElementById('edit-button-personal').style.display = 'none';

document.getElementById('save-button-personal').style.display = 'block';

document.getElementById('cancel-button-personal').style.display = 'block';
});

document.getElementById('cancel-button-personal').addEventListener('click',
function() {
    restoreOriginalValues(); // Restaurar los valores
    originales al cancelar la edición
    var fields =
document.querySelectorAll('#personal-info-form .form-control');
    fields.forEach(function(field) {
        field.readOnly = true;
        field.disabled = true;
    });
}

document.getElementById('edit-button-personal').style.display = 'block';

document.getElementById('save-button-personal').style.display = 'none';

document.getElementById('cancel-button-personal').style.display = 'none';
    location.reload();
});
</script>

<?php Pjax::end(); ?>
```

- informacion\_laboral.

Descripción. Código que tiene la funcionalidad de mostrar la información laboral del empleado. Se encuentran permisos que permiten modificar o añadir la información mediante formularios.

```
<?php
//IMPORTAR
use app\models\CatDepartamento;
use app\models\CatPuesto;
use app\models\CatTipoContrato;
```

```
use yii\helpers\Html;
use kartik\form\ActiveForm;
use yii\helpers\ArrayHelper;
use app\models\JuntaGobierno;
use kartik\select2>Select2;

?>

<br>

<?php
//CARGAR REGISTROS DE EMPLEADOS QUE SEAN DIRECTORES,
JEFES DE UNIDAD O DEPARTAMENTO
$juntaDirectorDireccion = JuntaGobierno::find()
    ->where(['nivel_jerarquico' => 'Director'])
    ->andWhere(['cat_direccion_id' =>
$model->informacionLaboral->cat_direccion_id])
    ->one();

$jefesDirectores = ArrayHelper::map(
    JuntaGobierno::find()
        ->where(['nivel_jerarquico' => 'Jefe de unidad'])
        ->orWhere(['nivel_jerarquico' => 'Jefe de departamento'])
        ->andWhere(['cat_direccion_id' =>
$model->informacionLaboral->cat_direccion_id])
        ->all(),
    'id',
    function ($model) {
        return $model->empleado->nombre . ' ' . $model->empleado->apellido;
    }
);

?>
<div class="card">
    <?php
        //FORMULARIO PARA REGISTRAR LA INFORMACION LABORAL DEL
        EMPLEADO
        $form = ActiveForm::begin([
            'action' => ['actualizar-informacion-laboral', 'id' => $model->id],
            'options' => ['id' => 'laboral-info-form']
        ]); ?>
```

```
<div id="loading-spinner-laboral" style="display: none;">
    <i class="fa fa-spinner fa-spin fa-2x" style="color: #000000;"></i>
    Procesando...
</div>

<div class="card-header bg-info text-white">
    <h3>Información Laboral</h3>
    <?php if (Yii::$app->user->can('modificar-informacion-empleados')) : ?>

        <button type="button" id="edit-button-laboral" class="btn btn-light float-right"><i class="fa fa-edit"></i></button>
        <button type="button" id="cancel-button-laboral" class="btn btn-danger float-right" style="display:none;"><i class="fa fa-times"></i></button>
        <?= Html::submitButton('<i class="fa fa-save"></i>', ['class' => 'btn btn-success float-right mr-3', 'id' => 'save-button-laboral', 'style' => 'display:none;']) ?>
        <?php endif; ?>
    </div>
    <div class="card-body">
        <div class="row">
            <div class="col-6 col-sm-6">

                <div class="form-group">

                    <label class="font-weight-bold">Días Laborales</label><br>
                    <div id="dias-laborales-display" class="alert alert-warning">
                        <p><strong>Días laborales actuales:</strong> <?= Html::encode(implode(', ', explode(',', $model->informacionLaboral->dias_laborales))) ?></p>
                    </div>
                    <div id="dias-laborales-checkboxes" style="display: none;">
                        <div class="form-check form-check-inline">
                            <?= Html::checkbox('InformacionLaboral[dias_laborales][]', in_array('Lunes', explode(',', $model->informacionLaboral->dias_laborales)), ['value' => 'Lunes', 'class' => 'form-check-input', 'id' => 'lunes']) ?>
                            <?= Html::label('Lunes', 'lunes', ['class' => 'form-check-label']) ?>
                        </div>
                        <div class="form-check form-check-inline">
                            <?= Html::checkbox('InformacionLaboral[dias_laborales][]', in_array('Martes', explode(',', $model->informacionLaboral->dias_laborales)), ['value' => 'Martes', 'class' => 'form-check-input', 'id' => 'martes']) ?>
                            <?= Html::label('Martes', 'martes', ['class' => 'form-check-label']) ?>
                        </div>
                    </div>
                </div>
            </div>
        </div>
    </div>
</div>
```

```
<div class="form-check form-check-inline">
    <?= Html::checkbox('InformacionLaboral[dias_laborales][]',
in_array('Miércoles', explode(',', $model->informacionLaboral->dias_laborales)), ['value' => 'Miércoles', 'class' => 'form-check-input', 'id' => 'miercoles']) ?>
        <?= Html::label('Miércoles', 'miercoles', ['class' => 'form-check-label']) ?>
    </div>
<div class="form-check form-check-inline">
    <?= Html::checkbox('InformacionLaboral[dias_laborales][]',
in_array('Jueves', explode(',', $model->informacionLaboral->dias_laborales)), ['value' => 'Jueves', 'class' => 'form-check-input', 'id' => 'jueves']) ?>
        <?= Html::label('Jueves', 'jueves', ['class' => 'form-check-label']) ?>
    </div>
<div class="form-check form-check-inline">
    <?= Html::checkbox('InformacionLaboral[dias_laborales][]',
in_array('Viernes', explode(',', $model->informacionLaboral->dias_laborales)), ['value' => 'Viernes', 'class' => 'form-check-input', 'id' => 'viernes']) ?>
        <?= Html::label('Viernes', 'viernes', ['class' => 'form-check-label']) ?>
    </div>
<div class="form-check form-check-inline">
    <?= Html::checkbox('InformacionLaboral[dias_laborales][]',
in_array('Sábado', explode(',', $model->informacionLaboral->dias_laborales)), ['value' => 'Sábado', 'class' => 'form-check-input', 'id' => 'sabado']) ?>
        <?= Html::label('Sábado', 'sabado', ['class' => 'form-check-label']) ?>
    </div>
<div class="form-check form-check-inline">
    <?= Html::checkbox('InformacionLaboral[dias_laborales][]',
in_array('Domingo', explode(',', $model->informacionLaboral->dias_laborales)), ['value' => 'Domingo', 'class' => 'form-check-input', 'id' => 'domingo']) ?>
        <?= Html::label('Domingo', 'domingo', ['class' => 'form-check-label']) ?>
    </div>
</div>
</div>
<div class="col-6 col-sm-2">
    <?= $form->field($model->informacionLaboral,
'horario_laboral_inicio')->input('time', [
```

```
'disabled' => true,
'title' => $model->informacionLaboral->horario_laboral_inicio,
])->label('Hora de entrada', [
'title' => $model->informacionLaboral->horario_laboral_inicio,
]) ?>
</div>

<div class="col-6 col-sm-2">
<?= $form->field($model->informacionLaboral,
'horario_laboral_fin')->input('time', [
'disabled' => true,
'title' => $model->informacionLaboral->horario_laboral_fin,
])->label('Hora de salida', [
'title' => $model->informacionLaboral->horario_laboral_fin,
]) ?>
</div>

<div class="w-100"></div>

<?php if
(Yii::$app->user->can('ver-informacion-completa-empleados')) : ?>
<div class="col-6 col-sm-2">
<?= $form->field($model->informacionLaboral,
'cat_tipo_contrato_id')->widget(Select2::classname(), [
'data' => ArrayHelper::map(CatTipoContrato::find()->all(), 'id',
'nombre_tipo'),
'options' => [
'placeholder' => 'Seleccionar Tipo de Contrato',
'disabled' => true,
'title' => $model->informacionLaboral->cat_tipo_contrato_id
?
CatTipoContrato::findOne($model->informacionLaboral->cat_tipo_contrato_id
)->nombre_tipo : '',
],
'pluginOptions' => [
'allowClear' => true
],
'theme' => Select2::THEME_KRAJEE_BS3,
])->label('Tipo de contrato', [
'title' => $model->informacionLaboral->cat_tipo_contrato_id ?
CatTipoContrato::findOne($model->informacionLaboral->cat_tipo_contrato_id
)->nombre_tipo : ''
]) ?>
```

```
</div>
<?php endif; ?>
<div class="col-6 col-sm-2">
    <?= $form->field($model->informacionLaboral,
    'fecha_ingreso')->input('date', [
        'disabled' => true,
        'title' => $model->informacionLaboral->fecha_ingreso,
    ])?->label('Fecha de ingreso', [
        'title' => $model->informacionLaboral->fecha_ingreso,
    ]) ?>
</div>

<div class="col-6 col-sm-4">
    <?= $form->field($model->informacionLaboral,
    'cat_departamento_id')->widget(Select2::classname(), [
        'data' => ArrayHelper::map(CatDepartamento::find()->all(), 'id',
    'nombre_departamento'),
        'options' => [
            'placeholder' => 'Seleccionar Departamento',
            'disabled' => true,
            'title' => $model->informacionLaboral->cat_departamento_id ?
CatDepartamento::findOne($model->informacionLaboral->cat_departamento
_id)->nombre_departamento : '',
        ],
        'pluginOptions' => [
            'allowClear' => true
        ],
        'theme' => Select2::THEME_KRAJEE_BS3,
    ])?->label('Departamento', ['title' =>
$model->informacionLaboral->cat_departamento_id ?

CatDepartamento::findOne($model->informacionLaboral->cat_departamento
_id)->nombre_departamento : "]) ?>
</div>

<div class="col-6 col-sm-2">
    <div class="form-group">
```

```
<label class="control-label" title="<?= Html::encode($model->informacionLaboral->catDireccion->nombredireccion); ?>">Dirección:</label>
<p id="edad-label" title="<?= Html::encode($model->informacionLaboral->catDireccion->nombredireccion); ?>><?= Html::encode($model->informacionLaboral->catDireccion->nombredireccion); ?></p>
</div>
</div>

<div class="col-6 col-sm-2">
<div class="form-group">
<label class="control-label" title="<?= Html::encode($model->informacionLaboral->catDptoCargo->nombredpto); ?>">DPTO:</label>
<p id="edad-label" title="<?= Html::encode($model->informacionLaboral->catDptoCargo->nombredpto); ?>><?= Html::encode($model->informacionLaboral->catDptoCargo->nombredpto); ?></p>
</div>
</div>

<div class="w-100"></div>

<div class="w-100"></div>

<div class="col-6 col-sm-3">
<?= $form->field($model->informacionLaboral, 'junta_gobierno_id')->widget(Select2::classname(), [
    'data' => $jefesDirectores,
    'options' => [
        'placeholder' => 'Seleccionar Jefe o director a cargo',
        'disabled' => true,
        'title' =>
isset($jefesDirectores[$model->informacionLaboral->junta_gobierno_id]) ?
$adosignado',
        ],
    'pluginOptions' => [
        'allowClear' => true
    ],
    'theme' => Select2::THEME_KRAJEE_BS3,
```

```
])->label('Jefe inmediato', ['title' =>
isset($jefesDirectores[$model->informacionLaboral->junta_gobierno_id]) ?  

  
$jefesDirectores[$model->informacionLaboral->junta_gobierno_id] : 'No  
asignado']) ?>
</div>  
  
<div class="col-6 col-sm-3">  
  
    <div class="form-group">
        <label class="control-label" title=<?= $juntaDirectorDireccion ?  
Html::encode($juntaDirectorDireccion->empleado->nombre . '' .
$juntaDirectorDireccion->empleado->apellido) : 'No Asignado' ?>>>Director  
de dirección</label>
        <p id="director-label" title=<?= $juntaDirectorDireccion ?  
Html::encode($juntaDirectorDireccion->empleado->nombre . '' .
$juntaDirectorDireccion->empleado->apellido) : 'No asignado' ?>><?=
$juntaDirectorDireccion ?
Html::encode($juntaDirectorDireccion->empleado->nombre . '' .
$juntaDirectorDireccion->empleado->apellido) : 'No asignado' ?>></p>
        </div>
    </div>  
  
    <div class="col-6 col-sm-4">
        <?= $form->field($model->informacionLaboral,
'cat_puesto_id')->widget(Select2::classname(), [
            'data' => ArrayHelper::map(CatPuesto::find()->all(), 'id',
'nombre_puesto'),
            'options' => [
                'placeholder' => 'Seleccionar Puesto',
                'disabled' => true,
                'title' => $model->informacionLaboral->cat_puesto_id ?  

  
CatPuesto::findOne($model->informacionLaboral->cat_puesto_id)->nombre_
puesto : '',
                ],
            'pluginOptions' => [
                'allowClear' => true
            ],
            'theme' => Select2::THEME_KRAJEE_BS3,  

  
])->label('Nombramiento', [
```

```
'title' => $model->informacionLaboral->cat_puesto_id ?  
  
CatPuesto::findOne($model->informacionLaboral->cat_puesto_id)->nOMBRE_  
puesto : "  
    ]) ?>  
</div>  
  
  
<?php if  
(Yii::$app->user->can('ver-informacion-completa-empleados')) : ?>  
    <div class="col-6 col-sm-4">  
        <?= $form->field($model->informacionLaboral,  
'numero_cuenta')->input('number', [  
            'maxlength' => 18,  
            'readonly' => true,  
            'class' => 'form-control no-spinner',  
            'title' => $model->informacionLaboral->numero_cuenta,  
        ])->label('Número de cuenta', [  
            'title' => $model->informacionLaboral->numero_cuenta  
        ]) ?>  
</div>  
  
  
<?php endif; ?>  
<?php if  
(Yii::$app->user->can('ver-informacion-completa-empleados')) : ?>  
  
    <div class="col-6 col-sm-3">  
        <?= $form->field($model->informacionLaboral,  
'salario')->textInput([  
            'type' => 'number',  
            'step' => '0.01',  
            'readonly' => true,  
            'class' => 'form-control',  
            'title' => $model->informacionLaboral->salario  
        ])->label('Salario', [  
            'title' => $model->informacionLaboral->salario  
        ]) ?>  
</div>  
  
  
<?php endif; ?>  
  
<?php ActiveForm::end(); ?>  
<?php  
$script = <<< JS
```

```
$(document).ready(function() {
    $('#laboral-info-form').on('beforeSubmit', function() {
        var button = $('#save-button-laboral');
        var spinner = $('#loading-spinner-laboral');

        button.prop('disabled', true); // Deshabilita el botón
        spinner.show(); // Muestra el spinner

        return true; // Permite que el formulario se envíe
    });
});
JS;
    $this->registerJs($script);
?>

</div>
</div>
</div>

<script>
    document.addEventListener("DOMContentLoaded", function() {
        // Al cargar la página, guardar los valores originales de los campos
        var originalValues = {};
        var fields = document.querySelectorAll('#laboral-info-form input,
#laboral-info-form select');
        fields.forEach(function(field) {
            originalValues[field.id] = field.value;
        });

        // Botón Editar
        document.getElementById('edit-button-laboral').addEventListener('click',
function() {
            fields.forEach(function(field) {
                field.readOnly = false;
                field.disabled = false;
            });
            document.getElementById('dias-laborales-display').style.display =
'none';
            document.getElementById('dias-laborales-checkboxes').style.display =
'block';
            document.getElementById('edit-button-laboral').style.display = 'none';
            document.getElementById('save-button-laboral').style.display =
'block';
        });
    });
}
```

```
document.getElementById('cancel-button-laboral').style.display =
'block';
});

// Botón Cancelar

document.getElementById('cancel-button-laboral').addEventListener('click',
function() {
    fields.forEach(function(field) {
        field.readOnly = true;
        field.disabled = true;
        // Restaurar al valor original
        field.value = originalValues[field.id];
    });
    document.getElementById('dias-laborales-display').style.display =
'block';
    document.getElementById('dias-laborales-checkboxes').style.display =
='none';
    document.getElementById('edit-button-laboral').style.display = 'block';
    document.getElementById('save-button-laboral').style.display = 'none';
    document.getElementById('cancel-button-laboral').style.display =
'none';

    location.reload();
});
});
}
</script>
```

- informacion\_vacacional.

Descripción. Código que tiene la funcionalidad de mostrar la información vacacional del empleado. Se encuentran permisos que permiten modificar o añadir la información mediante formularios.

```
<?php
//IMPORTACIONES
use kartik\form\ActiveForm;
use yii\helpers\Html;
use yii\web\JsExpression;
use kartik\daterange\DateRangePicker;

?>
```

```
<div class="card">
```

```
<div class="card-body">
    <div class="card-header bg-success text-dark text-center">
        <h3>Total de dias vacacionales: <?= $model->informacionLaboral->vacaciones->total_dias_vacaciones ?></h3>
    </div>

    <li class="dropdown-divider"></li>

    <div class="row">
        <div class="col-md-6">
            <div class="card">
                <?php
                    //FORMULARIO PARA REGISTRAR LA
                    INFORMACIÓN VACACIONAL DEL EMPLEADO
                    $form = ActiveForm::begin([
                        'action' => ['actualizar-primer-periodo', 'id' => $model->id],
                        'options' => ['id' => 'first-period-form']
                    ]); ?>
                <div class="card-header bg-info text-white">
                    <h3>PRIMER PERIODO</h3>
                    <?php setlocale(LC_TIME, "es_419.UTF-8"); ?>
                    <div class="alert alert-warning text-center"
role="alert">
                        <label class="control-label small">
                            <?php if
($model->informacionLaboral->vacaciones->periodoVacacional &&
$model->informacionLaboral->vacaciones->periodoVacacional->fecha_inicio &&
$model->informacionLaboral->vacaciones->periodoVacacional->fecha_final) : ?>
                                <?= mb_strtoupper(strftime('%A, %d de %B
de %Y',
strtotime($model->informacionLaboral->vacaciones->periodoVacacional->fecha_inicio))) ?>
                                ---
                                <?= mb_strtoupper(strftime('%A, %d de %B
de %Y',
strtotime($model->informacionLaboral->vacaciones->periodoVacacional->fecha_final))) ?>
                            <?php else : ?>
                                Aún no se ha definido el periodo
                            <?php endif; ?>
                        </label>
                    </div>
                </div>
            </div>
        </div>
    </div>
</div>
```

```
</div>
<label>Dias de vacaciones disponibles: <?=
$model->informacionLaboral->vacaciones->periodoVacacional->dias_vacaciones_p
eriodo ?></label><br>
<label>Dias de vacaciones restantes: <span
id="dias-disponibles"><?=
$model->informacionLaboral->vacaciones->periodoVacacional->dias_disponibles
?></span></label><br>
<?php if
(Yii::$app->user->can('modificar-informacion-empleados')) : ?>

<button type="button" id="edit-button-first-period"
class="btn btn-light float-right"><i class="fa fa-edit"></i></button>
<button type="button"
id="cancel-button-first-period" class="btn btn-danger float-right"
style="display:none;"><i class="fa fa-times"></i></button>

<?= Html::submitButton('<i class="fa
fa-save"></i>', ['class' => 'btn btn-dark float-right mr-3', 'id' =>
'save-button-first-period', 'style' => 'display:none;']) ?>
<?php endif; ?>
</div>
<div class="card-body">
<?=
$form->field($model->informacionLaboral->vacaciones->periodoVacacional,
'año')->textInput(['type' => 'number', 'disabled' => true, 'autocomplete'=>"off",]) ?>
<?=
$form->field($model->informacionLaboral->vacaciones->periodoVacacional,
'dateRange')->widget(DateRangePicker::class, [
    'convertFormat' => true,
    'pluginOptions' => [
        'locale' => [
            'format' => 'Y-m-d',
            'separator' => ' a ',
        ],
        'opens' => 'left',
        'singleDatePicker' => false,
        'showDropdowns' => true,
        'alwaysShowCalendars' => true,
        'minDate' => '2000-01-01',
        'maxDate' => '2100-12-31',
        'startDate' => $currentDate,
```

```
'endDate' => $currentDate,  
    'autoApply' => true,  
],  
    'options' => ['disabled' =>  
true,'autocomplete'=>"off",  
],  
    'pluginEvents' => [  
        "apply.daterangepicker" => new  
JsExpression("function(ev, picker) {  
    var startDate = picker.startDate.format('YYYY-MM-DD');  
    var endDate = picker.endDate.format('YYYY-MM-DD');  
    var diasSeleccionados = picker.endDate.diff(picker.startDate,  
'days') + 1;  
  
    if (diasSeleccionados >  
{$model->informacionLaboral->vacaciones->periodoVacacional->dias_vacaciones_p  
eriodo}) {  
        alert('No puede seleccionar un rango de fechas que exceda  
los días disponibles.');//  
        picker.startDate = picker.oldStartDate;  
        picker.endDate = picker.oldEndDate;  
        picker.updateView();  
        picker.renderCalendar();  
  
        $('#first-period-form').data('daterangepicker').setStartDate(picker.oldStartDate);  
  
        $('#first-period-form').data('daterangepicker').setEndDate(picker.oldEndDate);  
    } else {  
        $('#dias-disponibles').text(diasDisponibles);  
    }  
},  
],  
])->label('Seleccionar rango de fechas del primer  
periodo:') ?>  
    <?=br/>$form->field($model->informacionLaboral->vacaciones->periodoVacacional,  
'original')->dropDownList(['Si' => 'Si', 'No' => 'No'], ['prompt' => 'Selecciona una  
opción...', 'disabled' => true]) ?>  
    </div>  
    <?php ActiveForm::end(); ?>  
    </div>  
    </div>
```

```
<script>
    //SCRIPT PARA HABILITAR LA EDICION DE LOS
CAMPOS

document.getElementById('edit-button-first-period').addEventListener('click',
function() {
    var fields =
document.querySelectorAll('#first-period-form input, #first-period-form select');
    fields.forEach(function(field) {
        field.disabled = false;
    });
    $('.select2-hidden-accessible').select2('enable');

document.getElementById('edit-button-first-period').style.display = 'none';

document.getElementById('save-button-first-period').style.display = 'block';

document.getElementById('cancel-button-first-period').style.display = 'block';
});

document.getElementById('cancel-button-first-period').addEventListener('click',
function() {
    var fields =
document.querySelectorAll('#first-period-form input, #first-period-form select');
    fields.forEach(function(field) {
        field.disabled = true;
        if (field.tagName !== 'SELECT') {
            field.value = field.defaultValue;
        }
    });
    $('.select2-hidden-accessible').select2('enable', false);

document.getElementById('edit-button-first-period').style.display = 'block';

document.getElementById('save-button-first-period').style.display = 'none';
});
```

```
document.getElementById('cancel-button-first-period').style.display = 'none';
    });
</script>

<div class="col-md-6">
    <div class="card">
        <?php $form = ActiveForm::begin([
            'action' => ['actualizar-segundo-periodo', 'id' =>
$model->id],
            'options' => ['id' => 'second-period-form']
        ]); ?>
        <div class="card-header bg-secondary text-white">
            <h3>SEGUNDO PERIODO</h3>
            <?php setlocale(LC_TIME, "es_419.UTF-8"); ?>
            <div class="alert alert-warning text-center"
role="alert">
                <label class="control-label small">
                    <?php if
($model->informacionLaboral->vacaciones->segundoPeriodoVacacional &&
$model->informacionLaboral->vacaciones->segundoPeriodoVacacional->fecha_inici
o &&
$model->informacionLaboral->vacaciones->segundoPeriodoVacacional->fecha_final
) : ?>
                        <?= mb_strtoupper(strftime('%A, %d de %B
de %Y',
strtotime($model->informacionLaboral->vacaciones->segundoPeriodoVacacional->fe
cha_inicio))) ?>
                        ---
                        <?= mb_strtoupper(strftime('%A, %d de %B
de %Y',
strtotime($model->informacionLaboral->vacaciones->segundoPeriodoVacacional->fe
cha_final))) ?>
                <?php else : ?>
                    Aún no se ha definido el periodo
                <?php endif; ?>
            </label>
        </div>
```

```
<label> Dias de vacaciones disponibles: <?= $model->informacionLaboral->vacaciones->segundoPeriodoVacacional->dias_vacaciones_periodo ?>
</label><br>
<label> Dias de vacaciones restantes: <?= $model->informacionLaboral->vacaciones->segundoPeriodoVacacional->dias_disponibles ?>
</label>

<br>
<?php if (Yii::$app->user->can('modificar-informacion-empleados')) : ?>

    <button type="button" id="edit-button-period" class="btn btn-light float-right"><i class="fa fa-edit"></i></button>
    <button type="button" id="cancel-button-period" class="btn btn-danger float-right" style="display:none;"><i class="fa fa-times"></i></button>
    <?= Html::submitButton('<i class="fa fa-save"></i>', ['class' => 'btn btn-info float-right mr-3', 'id' => 'save-button-period', 'style' => 'display:none;']) ?>
    <?php endif; ?>

</div>
<div class="card-body">
    <?= $form->field($model->informacionLaboral->vacaciones->segundoPeriodoVacacional, 'año')->textInput(['type' => 'number', 'disabled' => true, 'autocomplete'=>"off",]) ?>
    <?= $form->field($model->informacionLaboral->vacaciones->segundoPeriodoVacacional, 'dateRange')->widget(DateRangePicker::class, [
        'convertFormat' => true,
        'pluginOptions' => [
            'locale' => [
                'format' => 'Y-m-d',
                'separator' => ' a ',
            ],
            'opens' => 'left',
            'singleDatePicker' => false,
            'showDropdowns' => true,
            'alwaysShowCalendars' => true,
            'minDate' => '2000-01-01',
        ],
    ]) ?>
```

```
'maxDate' => '2100-12-31',
'startDate' => $currentDate,
'endDate' => $currentDate,
'autoApply' => true,
],
'options' => ['disabled' => true,
'autocomplete'=>"off",],
'pluginEvents' => [
    "apply.daterangepicker" => new
JsExpression("function(ev, picker) {
    var startDate =
picker.startDate.format('YYYY-MM-DD');
    var endDate =
picker.endDate.format('YYYY-MM-DD');
    var diasSeleccionados =
picker.endDate.diff(picker.startDate, 'days') + 1;

    if (diasSeleccionados >
{$model->}informacionLaboral->}vacaciones->}segundoPeriodoVacacional->}dias_vaca
ciones_periodo) {
        alert('No puede seleccionar un rango de
fechas que exceda los días disponibles.');
        picker.startDate = picker.oldStartDate;
        picker.endDate = picker.oldEndDate;
        picker.updateView();
        picker.renderCalendar();

        $('#first-period-form').data('daterangepicker').setStartDate(picker.oldStartDate);

        $('#first-period-form').data('daterangepicker').setEndDate(picker.oldEndDate);
    } else {
        $('#dias-disponibles').text(diasDisponibles);
    }
}),
],
])->label('Seleccionar rango de fechas del segundo
periodo:') ?>
<?=
$form->field($model->}informacionLaboral->}vacaciones->}segundoPeriodoVacacional,
```

```
'original')->dropDownList(['Si' => 'Si', 'No' => 'No'], ['prompt' => 'Selecciona una opción...', 'disabled' => true]) ?>
    </div>
    <?php ActiveForm::end(); ?>
</div>

</div>
<script>
//SCRIPT PARA HABILITAR LA EDICION DE LOS
CAMPOS

document.getElementById('edit-button-period').addEventListener('click', function() {
    var fields =
document.querySelectorAll('#second-period-form input, #second-period-form
select');
    fields.forEach(function(field) {
        field.disabled = false;
    });
}

document.getElementById('edit-button-period').style.display = 'none';

document.getElementById('save-button-period').style.display = 'block';

document.getElementById('cancel-button-period').style.display = 'block';
});

document.getElementById('cancel-button-period').addEventListener('click', function()
{
    var fields =
document.querySelectorAll('#second-period-form input, #second-period-form
select');
    fields.forEach(function(field) {
        field.disabled = true;
        if (!field.type === 'select-one') {
            field.value = field.defaultValue;
        }
    });
}

document.getElementById('edit-button-period').style.display = 'block';
```

```
document.getElementById('save-button-period').style.display = 'none';

document.getElementById('cancel-button-period').style.display = 'none';
    });
</script>

<div class="col-md-6">
    <div class="card">
        <div class="card-header bg-success text-white">
            <h3>Historial de cambios de Periodos</h3>
            <button type="button" id="toggle-historial-button"
class="btn btn-light float-right"><i class="fa fa-eye"></i> Mostrar Historial</button>
        </div>
        <div class="card-body" id="historial-content"
style="display: none;">
            <ul class="list-group">
                <?php foreach ($historial as $item) : ?>
                <li class="list-group-item">
                    <strong><?= ucfirst($item->periodo)
?></strong><br>
                    <?= mb_strtoupper(strftime('Fecha inicio:
%A, %d de %B de %Y', strtotime($item->fecha_inicio))) ?><br>
                    <?= mb_strtoupper(strftime('Fecha final:
%A, %d de %B de %Y', strtotime($item->fecha_final))) ?><br>
                
            
            Año: <?= $item->año ?><br>
            Días Disponibles: <?=
            $item->dias_disponibles ?><br>
            Original: <?= $item->original ?><br>
            </li>
        <?php endforeach; ?>
    </ul>
    </div>
</div>

<script>
```

### //SCRIPT PARA MOSTRAR EL HISTORIAL DE CAMBIOS EN LA INFORMACIÓN VACACIONAL

```
document.getElementById('toggle-historial-button').addEventListener('click',
function() {
    var historialContent =
document.getElementById('historial-content');
    if (historialContent.style.display === 'none') {
        historialContent.style.display = 'block';
        this.innerHTML = '<i class="fa fa-eye-slash"></i>
Ocultar Historial';
    } else {
        historialContent.style.display = 'none';
        this.innerHTML = '<i class="fa fa-eye"></i> Mostrar
Historial';
    }
});
```

</script>

</div>

</div>

</div>

- informacion\_solicitudes.

Descripción. Código que tiene la funcionalidad de mostrar los registros o el historial de solicitudes realizadas por el empleado. Esta lista que muestra las solicitudes de todo tipo, sin embargo, para el rol médico solo muestra las solicitudes de citas médicas que ha realizado el empleado.

```
<?php
// IMPORTACIONES
use yii\helpers\Html;

use yii\grid\GridView;
use yii\widgets\Pjax;
use yii\helpers\Url;
use yii\jui\DatePicker;

use kartik\select2\Select2;
```

```
?>
<div class="card">
    <div class="card-body">
        <div class="card-header bg-success text-dark text-center">

            <?php
            //PERMISO QUE MUESTRA TODAS LAS SOLICITUDES
            if (Yii::$app->user->can('ver-solicitudes-formatos')) { ?>

                <h3>HISTORIAL DE SOLICITUDES DE INCIDENCIAS:
            </h3>

                <div class="dropdown float-right">
                    <button class="btn btn-dark dropdown-toggle" type="button"
data-toggle="dropdown" aria-expanded="false">
                        FORMATOS DE INCIDENCIAS
                    </button>
                    <div class="dropdown-menu">
                        <?= Html::a('CITA MEDICA', Url::to(['cita-medica/historial', 'empleado_id' =>
$model->id]), ['class' => 'dropdown-item text-primary', 'data-pjax' => '0']) ?>

                        <?= Html::a('PERMISO FUERA DEL TRABAJO',
Url::to(['permiso-fuera-trabajo/historial', 'empleado_id' => $model->id]), ['class' =>
'dropdown-item text-primary', 'data-pjax' => '0']) ?>

                        <?= Html::a('COMISIÓN ESPECIAL', Url::to(['comision-especial/historial',
'empleado_id' => $model->id]), ['class' => 'dropdown-item text-primary', 'data-pjax'
=> '0']) ?>

                        <?= Html::a('CAMBIO DE DÍA LABORAL', Url::to(['cambio-dia-laboral/historial',
'empleado_id' => $model->id]), ['class' => 'dropdown-item text-primary', 'data-pjax'
=> '0']) ?>

                        <?= Html::a('CAMBIO DE HORARIO DE TRABAJO',
Url::to(['cambio-horario-trabajo/historial', 'empleado_id' => $model->id]), ['class' =>
'dropdown-item text-primary', 'data-pjax' => '0']) ?>

                        <?= Html::a('PERMISO ECONÓMICO', Url::to(['permiso-economico/historial',
'empleado_id' => $model->id]), ['class' => 'dropdown-item text-primary', 'data-pjax'
=> '0']) ?>
```

```
<?= Html::a('PERMISO SIN GOCE DE SUELDO',
Url::to(['permiso-sin-sueldo/historial', 'empleado_id' => $model->id]), ['class' =>
'dropdown-item text-primary', 'data-pjax' => '0']) ?>
<?= Html::a('CAMBIO DE PERIODO VACACIONAL',
Url::to(['cambio-periodo-vacacional/historial', 'empleado_id' => $model->id]), ['class' =>
'dropdown-item text-primary', 'data-pjax' => '0']) ?>
<?= Html::a('REPORTE DE TIEMPO EXTRA',
Url::to(['reporte-tiempo-extra/historial', 'empleado_id' => $model->id]), ['class' =>
'dropdown-item text-primary', 'data-pjax' => '0'])?>
<?= Html::a('REPORTE DE TIEMPO EXTRA GENERAL',
Url::to(['reporte-tiempo-extra-general/index']), ['class' => 'dropdown-item
text-primary']) ?>
```

```
</div>
</div>
```

```
<?php } elseif
(Yii::$app->user->can('ver-solicitudes-medicas')) { ?>
```

```
<h3>HISTORIAL DE SOLICITUDES MEDICAS: </h3>
```

```
<?php } ?>
</div>
```

```
<li class="dropdown-divider"></li>
```

```
<?php
//PERMISO QUE MUESTRA SOLO EL HISTORIAL DE
SOLICITUDES MEDICAS
if (Yii::$app->user->can('solicitudes-medicas-view-medico')) {
?>
<div class="row">
<?php Pjax::begin(); ?>
<?= GridView::widget([
'dataProvider' => $dataProvider,
'filterModel' => $searchModel,
'columns' => [
['class' => 'yii\grid\SerialColumn'],

```

```
[  
    'class' => 'yii\grid\DataColumn',  
    'header' => "",  
    'format' => 'raw',  
    'value' => function ($model) { // MUESTRA EL ESTATUS DE SOLICITUD  
        PARA INDENTIFICAR SI ESTA ES NUEVA  
        if ($model->status === 'Nueva') {  
            return '<i class="fa fa-fire" aria-hidden="true" style="color: #ea4242  
"></i>;  
        } elseif ($model->status === 'Visto') {  
            return '<i class="fa fa-check-double" aria-hidden="true" style="color:  
#4678fc"></i>;  
  
        } else {  
            return '<i class="fa fa-question" aria-hidden="true" style="color:  
#6c757d"></i>;  
        }  
    },  
    'contentOptions' => ['class' => 'text-center'],  
    'headerOptions' => ['class' => 'text-center'],  
],  
[  
    'attribute' => 'empleado_id',  
    'label' => 'Empleado',  
    'value' => function ($model) {  
        return $model->empleado ? $model->empleado->nombre . ' ' .  
        $model->empleado->apellido : 'N/A';  
    },  
    'filter' => false,  
],  
[  
    'attribute' => 'fecha_creacion',  
    'format' => 'raw',  
    'value' => function ($model) {  
        setlocale(LC_TIME, "es_419.UTF-8");  
        return strftime('%A, %d de %B de %Y',  
        strtotime($model->fecha_creacion));  
    },  
    'filter' => DatePicker::widget([  
        'model' => $searchModel,  
        'attribute' => 'fecha_creacion',  
        'language' => 'es',  
    ]),  
]
```

```
'dateFormat' => 'php:Y-m-d',
'options' => [
    'class' => 'form-control',
    'autocomplete' => 'off',
],
'clientOptions' => [
    'changeYear' => true,
    'changeMonth' => true,
    'yearRange' => '-100:+0',
],
]),
],
[
    'class' => 'hail812\adminlte3\yii\grid\ActionColumn',
    'template' => '{view} {delete}',
    'buttons' => [
        'view' => function ($url, $model, $key) {
            return Html::a('<i class="fa fa-eye"></i>', ['solicitud/view', 'id' => $model->id], ['title' => 'Ver', 'class' => 'btn btn-outline-info btn-sm']);
        },
        'delete' => function ($url, $model, $key) {
            return Html::a('<i class="fa fa-trash"></i>', ['solicitud/delete', 'id' => $model->id], [
                'title' => 'Eliminar',
                'class' => 'btn btn-danger btn-xs',
                'data-confirm' => '¿Estás seguro de eliminar este elemento?',
                'data-method' => 'post',
            ]);
        },
    ],
],
],
'summaryOptions' => ['class' => 'summary mb-2'],
'pager' => [
    'class' => 'yii\bootstrap4\LinkPager',
],
]);
)?> <?php Pjax::end(); ?>
```

</div>

```
<?php } elseif (Yii::$app->user->can('solicitudes-medicas-view-empleado')) {  
//MUESTRA TODO TIPO DE SOLICITUDES?>  
<div class="row">  
<?php Pjax::begin(); ?>  
<?= GridView::widget([  
    'dataProvider' => $dataProvider,  
    'filterModel' => $searchModel,  
    'columns' => [  
        'class' => 'yii\grid\SerialColumn'],  
  
        [  
            'attribute' => 'empleado_id',  
            'label' => 'Empleado',  
            'value' => function ($model) {  
                return $model->empleado ? $model->empleado->nombre . ' '.  
$model->empleado->apellido : 'N/A';  
            },  
            'filter' => false,  
        ],  
        [  
            'attribute' => 'fecha_creacion',  
            'format' => 'raw',  
            'value' => function ($model) {  
                setlocale(LC_TIME, "es_419.UTF-8");  
                return strftime('%A, %d de %B de %Y',  
strtoftime($model->fecha_creacion));  
            },  
            'filter' => DatePicker::widget([  
                'model' => $searchModel,  
                'attribute' => 'fecha_creacion',  
                'language' => 'es',  
                'dateFormat' => 'php:Y-m-d',  
                'options' => [  
                    'class' => 'form-control',  
                    'autocomplete' => 'off',  
                ],  
                'clientOptions' => [  
                    'changeYear' => true,  
                    'changeMonth' => true,  
                    'yearRange' => '-100:+0',  
                ],  
            ]),  
        ]),
```

```
],
[
    'attribute' => 'nombre_formato',
    'label' => 'Tipo de solicitud',
    'value' => function ($model) {
        return $model->nombre_formato;
    },
    'filter' => Select2::widget([
        'model' => $searchModel,
        'attribute' => 'nombre_formato',
        'data' =>
\yii\helpers\ArrayHelper::map(\app\models\Solicitud::find()->select(['nombre_formato'])
)->distinct()->all(), 'nombre_formato', 'nombre_formato'),
        'options' => ['placeholder' => 'Seleccione un tipo de solicitud'],
        'pluginOptions' => [
            'allowClear' => true
        ],
        'theme' => Select2::THEME_KRAJEE_BS3,
    ]),
],
[
    'class' => 'hail812\adminlte3\yii\grid\ActionColumn',
    'template' => '{view}',
    'buttons' => [
        'view' => function ($url, $model, $key) {
            return Html::a('<i class="fa fa-eye"></i>', ['solicitud/view', 'id' =>
$model->id], ['title' => 'Ver', 'class' => 'btn btn-outline-info btn-sm']);
        },
    ],
],
['summaryOptions' => ['class' => 'summary mb-2'],
'pager' => [
    'class' => 'yii\bootstrap4\LinkPager',
],
]);
?>
<?php Pjax::end(); ?>

</div>

<?php } elseif (Yii::$app->user->can('ver-solicitudes-formatos')) { ?>
<div class="row">
```

```
<?php Pjax::begin(); ?>
<?= GridView::widget([
    'dataProvider' => $dataProvider,
    'filterModel' => $searchModel,
    'columns' => [
        ['class' => 'yii\grid\SerialColumn'],
        [
            'class' => 'yii\grid\DataColumn',
            'header' => '',
            'format' => 'raw',

            'value' => function ($model) {
                if ($model->status === 'Nueva') {
                    return '<i class="fa fa-fire" aria-hidden="true" style="color: #ea4242"></i>';
                } elseif ($model->status === 'Visto') {
                    return '<i class="fa fa-check-double" aria-hidden="true" style="color: #4678fc"></i>';
                } else {
                    return '<i class="fa fa-question" aria-hidden="true" style="color: #6c757d"></i>';
                }
            },
            'contentOptions' => ['class' => 'text-center'],
            'headerOptions' => ['class' => 'text-center'],
        ],
        [
            'attribute' => 'empleado_id',
            'label' => 'Empleado',
            'value' => function ($model) {
                return $model->empleado ? $model->empleado->nombre . ' ' .
$model->empleado->apellido : 'N/A';
            },
            'filter' => false,
        ],
        [
            'attribute' => 'fecha_creacion',
            'format' => 'raw',
            'value' => function ($model) {
                setlocale(LC_TIME, "es_419.UTF-8");
            }
        ]
    ]
]);
```

```
return strftime('%A, %d de %B de %Y',
strtotime($model->fecha_creacion));
},
'filter' => DatePicker::widget([
    'model' => $searchModel,
    'attribute' => 'fecha_creacion',
    'language' => 'es',
    'dateFormat' => 'php:Y-m-d',
    'options' => [
        'class' => 'form-control',
        'autocomplete' => 'off',
    ],
    'clientOptions' => [
        'changeYear' => true,
        'changeMonth' => true,
        'yearRange' => '-100:+0',
    ],
]),
],
[
    'attribute' => 'nombre_formato',
    'label' => 'Tipo de solicitud',
    'value' => function ($model) {
        return $model->nombre_formato;
    },
    'filter' => Select2::widget([
        'model' => $searchModel,
        'attribute' => 'nombre_formato',
        'data' =>
            \yii\helpers\ArrayHelper::map(\app\models\Solicitud::find()->select(['nombre_formato'])
                ->distinct()->all(), 'nombre_formato', 'nombre_formato'),
        'options' => ['placeholder' => 'Seleccione un tipo de solicitud'],
        'pluginOptions' => [
            'allowClear' => true
        ],
        'theme' => Select2::THEME_KRAJEE_BS3,
    ]),
],
[
    'class' => 'hail812\adminlte3\yii\grid\ActionColumn',
    'template' => '{view}://{delete}',
    'buttons' => [

```

```
'view' => function ($url, $model, $key) {
    return Html::a('<i class="fa fa-eye"></i>', ['solicitud/view', 'id' =>
$model->id], ['title' => 'Ver', 'class' => 'btn btn-outline-info btn-sm']);
},
],
],
],
'summaryOptions' => ['class' => 'summary mb-2'],
'pager' => [
    'class' => 'yii\bootstrap4\LinkPager',
],
]);
?>
<?php Pjax::end(); ?>
</div>
<?php } ?>

</div>
</div>
```

- informacion\_consultas.

Descripción. Código que tiene la funcionalidad de mostrar los registros o el historial de consultas médicas que han sido realizadas al empleado por parte del médico.

```
<?php
//IMPORTACIONES
use yii\helpers\Html;
use yii\grid\GridView;
use yii\widgets\Pjax;
use yii\jui\DatePicker;

?>
<div class="card">
    <div class="card-body">
        <div class="card-header bg-success text-dark text-center">
            <h3>HISTORIAL DE CONSULTAS MEDICAS: </h3>
        </div>
    </div>
```

```
<li class="dropdown-divider"></li>
<div class="row">
    <?php Pjax::begin(); //SE INICIALIZA EL
AJAX O PJAX ?>
    <?= GridView::widget([
        'dataProvider' => $dataProviderConsultas,
        // SE EXTRAN LOS DATOS DE LOS REGISTROS
        CORRESPONDIENTES AL MODELO
        'filterModel' => $searchModelConsultas,
        //SE UTILIZA EL MODELO QUE SE ENCARGA DE BUSCAR Y
        FILTRAR
        'columns' => [
            ['class' => 'yii\grid\SerialColumn'],
            [
                'label' => 'Fecha de la consulta',
                'attribute' => 'created_at',
                'format' => 'raw',
                'value' => function ($model) {
                    setlocale(LC_TIME,
                    "es_419.UTF-8");
                    return strftime('%A, %d de %B de
                    %Y', strtotime($model->created_at));
                },
                'filter' => DatePicker::widget([
                    'model' => $searchModelConsultas,
                    'attribute' => 'created_at',
                    'language' => 'es',
                    'dateFormat' => 'php:Y-m-d',
                    'options' => [
                        'class' => 'form-control',
                        'autocomplete' => 'off',
                    ],
                    'clientOptions' => [
                        'changeYear' => true,
                        'changeMonth' => true,
                        'yearRange' => '-100:+0',
                    ],
                ]),
            ],
        ],
    ]);
}
```

```
[  
    'label' => 'Motivo de la consulta',  
    'attribute' => 'motivo',  
    'format' => 'html',  
    'value' => function ($model) {  
        return  
            \yii\helpers\Html::decode($model->motivo);  
    },  
    'filter' => false,  
    'options' => ['style' => 'width: 65%;'],  
],  
//ACCIONES QUE SE PUEDE REALIZAR  
EN LAS FILAS DEL GRIDVIEW  
[  
    'class' =>  
        'hail812\adminlte3\yii\grid\ActionColumn',  
        'template' => '{view} {delete}',  
        'buttons' => [  
            'view' => function ($url, $model,  
$key) {  
                return Html::a('<i class="fa  
fa-eye"></i>', ['consulta-medica/view', 'id' => $model->id], ['title' =>  
'Ver', 'class' => 'btn btn-primary btn-xs']);  
            },  
            'delete' => function ($url, $model,  
$key) {  
                return Html::a('<i class="fa  
fa-trash"></i>', ['consulta-medica/delete', 'id' => $model->id], [  
                    'title' => 'Eliminar',  
                    'class' => 'btn btn-danger  
btn-xs',  
                    'data-confirm' => '¿Estás  
seguro de eliminar este elemento?',  
                    'data-method' => 'post',  
                ]);  
            },  
        ],  
        'summaryOptions' => ['class' => 'summary  
mb-2'],  
        'pager' => [
```

```
'class' => 'yii\bootstrap4\LinkPager',
],
]); ?>
<?php Pjax::end(); ?>
</div>
</div>
</div>
```

- documentos\_empleado.

Descripción. Código que tiene la funcionalidad de mostrar los registros de los documentos que se han subido del empleado. Desde aquí se pueden descargar o mostrar. Este mismo código también se encarga de mediante el formulario, subir el registro de nuevos documentos que se quieran adjuntar al empleado.

```
<?php
//IMPORTACIONES
use yii\helpers\Html;
use froala\froalaeditor\FroalaEditorWidget;
use kartik\file\FileInput;
use yii\grid\GridView;
use yii\widgets\Pjax;
use kartik\form\ActiveForm;
use yii\jui\DatePicker;
use yii\helpers\ArrayHelper;
use app\models\DocumentoSearch;
use kartik\select2\Select2;
use app\models\CatTipoDocumento;
?>
<div class="row">

<div class="col-md-12">
    <div class="card">
        <div class="card-header bg-info text-white">
            <h3>Documentos del empleado</h3>
            <button type="button" id="toggle-expediente-button" class="btn btn-dark float-right">
                <i class="fa fa-upload"></i>&ampnbsp &ampnbsp Agregar nuevo
                archivo
            </button>
```

</div>

```
<script>
    //SCRIPT PARA MANEJAR EL BOTON QUE OCULTA Y
    MUESTRA EL FORMULARIO DE SUBIDA DE DOCUMENTO

    document.getElementById('toggle-expediente-button').addEventListener('click', function() {
        var expedienteContent =
            document.getElementById('expediente-content');
        if (expedienteContent.style.display === 'none') {
            expedienteContent.style.display = 'block';
            this.innerHTML = '<i class="fa fa-ban"></i> &nbsp;
Cancelar';
        } else {
            expedienteContent.style.display = 'none';
            this.innerHTML = '<i class="fa fa-upload"></i> &nbsp;
&nbsp; Agregar nuevo archivo';
        }
    });
</script>

<div class="card-body" id="expediente-content" style="display:
none;">

    <?php
    //FORMULARIO PARA SUBIR LOS DOCUMENTOS

    $form = ActiveForm::begin([
        'action' => ['documento/create', 'empleado_id' =>
$model->id],
        'options' => ['enctype' => 'multipart/form-data', 'class' =>
'narrow-form']
    ]); ?>

    <div class="form-group">
        <?= $form->field($documentoModel,
'cat_tipo_documento_id')->widget(Select2::classname(), [
        'data' =>
ArrayHelper::map(CatTipoDocumento::find()->all(), 'id', 'nombre_tipo'),
```

```
'language' => 'es',
'options' => ['placeholder' => 'Seleccione el tipo de
documento', 'id' => 'tipo-documento'],
'pluginOptions' => [
  'allowClear' => true
],
'theme' => Select2::THEME_BOOTSTRAP,
'pluginEvents' => [
  'select2:opening' => "function() {
$('.select2-selection__clear').html('<span class=\"fas
fa-times\"></span>'); }",
  'select2:opening' => "function() {
$('.select2-selection__clear').css('margin-left', '0px'); }",
],
])->label('Tipo de Documento') ?>
</div>

<div class="form-group">
<?= $form->field($documentoModel, 'nombre')->textInput([
  'maxlength' => true,
  'id' => 'nombre-archivo',
  'style' => 'display:none',
  'placeholder' => 'Ingrese el nombre del documento'
])->label(false) ?>
</div>
<div class="form-group">

<?= $form->field($documentoModel,
'observacion')->widget(FroalaEditorWidget::className(), [
  'clientOptions' => [
    'toolbarInline' => false,
    'theme' => 'royal', // optional: dark, red, gray, royal
    'language' => 'es', // optional: ar, bs, cs, da, de, en_ca,
    en_gb, en_us ...
    'height' => 150,
    'pluginsEnabled' => [

```

```
'align', 'charCounter', 'codeBeautifier', 'codeView',
'colors',
'draggable', 'entities', 'fontFamily',
'fontSize', 'fullscreen', 'inlineStyle',
'lineBreaker', 'link', 'lists', 'paragraphFormat',
'paragraphStyle',
'quickInsert', 'quote', 'save', 'table', 'url', 'wordPaste'
]
]
])
)?>

</div>

<div class="form-group">
<?= $form->field($documentoModel,
'ruta')->widget(FilePath::classname(), [
'options' => ['accept' => 'file/*'],
'pluginEvents' => [
'fileclear' => "function() {
$('#nombre-archivo').val('');
$('#tipo-archivo').val('');
}",
],
'pluginOptions' => [
'showUpload' => false,
'showCancel' => false,
],
])->label('Archivo') ?>
</div>

<div class="form-group">
<?= Html::submitButton('Subir archivo <i class="fa fa-upload"></i>', ['class' => 'btn btn-warning float-right', 'id' => 'save-button-personal']) ?>
</div>

<?php ActiveForm::end(); ?>
</div>
</div>
</div>
```

```
<div class="col-md-12 mt-3">
    <div class="card">

        <div class="card-body">
            <?php
//GRIDVIEW PARA MOSTRAR LOS REGISTROS DE LOS
DOCUMENTOS DE LOS EMPLEADOS

            $searchModel = new DocumentoSearch();
            $params = Yii::$app->request->queryParams;
            $params[$searchModel->formName()]['empleado_id'] =
$model->id;
            $dataProvider = $searchModel->search($params);
            ?>

            <?php Pjax::begin(); ?>

<?= GridView::widget([
    'dataProvider' => $dataProvider,
    'filterModel' => $searchModel,
    'columns' => [
        ['class' => 'yii\grid\SerialColumn'],
        [
            'attribute' => 'nombre',
            'value' => 'nombre',
            'filter' => false,
            'options' => ['style' => 'width: 30%;'],
        ],
        [
            'attribute' => 'fecha_subida',
            'format' => 'raw',
            'value' => function ($model) {
                setlocale(LC_TIME, "es_419.UTF-8");
                return strftime('%A, %d de %B de %Y',
strtoftime($model->fecha_subida));
            },
            'filter' => DatePicker::widget([
                'model' => $searchModel,
                'attribute' => 'fecha_subida',
                'language' => 'es',
            ])
        ]
    ]
]);
```

```
'dateFormat' => 'php:Y-m-d',
'options' => [
    'class' => 'form-control',
    'autocomplete' => 'off',
],
'clientOptions' => [
    'changeYear' => true,
    'changeMonth' => true,
    'yearRange' => '-100:+0',
],
]),
'options' => ['style' => 'width: 30%;'],
],
[
    'attribute' => 'observacion',
    'format' => 'html',
    'value' => function ($model) {
        return
        \yii\helpers\Html::decode($model->observacion);
    },
    'filter' => false,
    'options' => ['style' => 'width: 30%;'],
],
[
    //ACCIONES PARA PERMITIR LA VISUALIZACION,
    DESCARGAR O ELIMINAR LOS DOCUMENTOS ASOCIADOS AL
    EMPLEADO
    'class' => 'hail812\adminlte3\yii\grid\ActionColumn',
    'template' => '{view} {delete} {download}',
    'buttons' => [
        'view' => function ($url, $model) {
            return Html::a('<i class="far fa-eye"></i>',
['documento/open', 'id' => $model->id], [
            'target' => '_blank',
            'title' => 'Ver archivo',
            'class' => 'btn btn-info btn-xs',
            'data-pjax' => "0"
        ]);
    },
    'delete' => function ($url, $model) {
```

```
return Html::a('<i class="fas fa-trash"></i>',
['documento/delete', 'id' => $model->id, 'empleado_id' =>
$model->empleado_id], [
    'title' => Yii::t('yii', 'Eliminar'),
    'data-confirm' => Yii::t('yii', '¿Estás seguro de
que deseas eliminar este elemento?'),
    'data-method' => 'post',
    'class' => 'btn btn-danger btn-xs',
]);
},
'download' => function ($url, $model) {
    return Html::a('<i class="fas fa-download"></i>',
['documento/download', 'id' => $model->id], [
        'title' => 'Descargar archivo',
        'class' => 'btn btn-success btn-xs',
        'data-pjax' => "0"
]);
},
],
],
],
],
'summaryOptions' => ['class' => 'summary mb-2'],
'pager' => [
    'class' => 'yii\bootstrap4\LinkPager',
],
'tableOptions' => ['class' => 'no-style-gridview'],
'rowOptions' => function ($model, $key, $index, $grid) {
    return ['class' => 'no-style-gridview'];
},
]);
?>

<?php Pjax::end(); ?>
</div>
</div>
</div>
</div>

<?php
//SCRIPT PARA MOSTRAR CAMPO PARA AGREGAR UN NEUVO
TIPO DE DOCUMENTO
$this->registerJs("
$('#tipo-documento').change(function(){
```

```
var tipoDocumentoid = $(this).val();
var nombreArchivoInput = $('#nombre-archivo');

// Obtener el nombre del tipo de documento seleccionado
var tipoDocumentoNombre = $('#tipo-documento
option:selected').text();

// Verificar si se seleccionó 'OTRO'
if (tipoDocumentoNombre == 'OTRO') {
// Mostrar el campo de nombre y limpiar su valor
nombreArchivoInput.show().val("").focus();
} else {
// Ocultar el campo de nombre y asignar el nombre del tipo de
documento seleccionado
nombreArchivoInput.hide().val(tipoDocumentoNombre);
}
});
");
?>
```

- [documento\\_medico\\_empleado.](#)

Descripción. Código que tiene la funcionalidad de mostrar los registros de los documentos médicos que se han subido del empleado. Desde aquí se pueden descargar o mostrar. Este mismo código también se encarga de mediante el formulario, subir el registro de nuevos documentos que se quieran adjuntar al empleado.

```
<?php
//IMPORTACIONES
use yii\helpers\Html;
use froala\froalaeditor\FroalaEditorWidget;
use kartik\file\FileInput;
use yii\grid\GridView;
use yii\widgets\Pjax;
use kartik\form\ActiveForm;
use yii\jui\DatePicker;

use app\models\DocumentoMedicoSearch;

?>
```

```
<div class="row">
    <div class="col-md-12">
        <div class="card">
            <div class="card-header bg-info text-white">
                <h3>Documentos médicos del
                    empleado</h3>
                <?php
                    //PERMISO PARA PODER REALIZAR
                    ACCIONES DE LOS REGISTROS DE LOS DOCUMENTOS
                    MEDICOS
                    if
                    (Yii::$app->user->can('acciones-documentos-medicos')) { ?>

                        <button type="button"
                            id="toggle-expediente-medico-button" class="btn btn-dark float-right">
                            <i class="fa fa-upload"></i>&ampnbsp
                            &ampnbsp Agregar nuevo archivo
                        </button>

                        <?php } ?>
                    </div>

                    <script>

document.getElementById('toggle-expediente-medico-button').addEventListener('click', function() {
    var expedienteMedicoContent =
        document.getElementById('expediente-medico-content');
    if
        (expedienteMedicoContent.style.display === 'none') {

            expedienteMedicoContent.style.display = 'block';
            this.innerHTML = '<i class="fa
            fa-ban"></i> &ampnbsp Cancelar';
        } else {

            expedienteMedicoContent.style.display = 'none';
            this.innerHTML = '<i class="fa
            fa-upload"></i> &ampnbsp Agregar nuevo archivo';
        }
    });
</script>
```

```
<div class="card-body"
id="expediente-medico-content" style="display: none;">
<?php
//FORMULARIO PARA SUBIR EL
ARCHIVO MEDICO Y CREAR UN NUEVO REGISTRO
$form = ActiveForm::begin([
    'action' => ['documento-medico/create'],
    'employee_id' => $model->id,
    'options' => ['enctype' =>
'multipart/form-data', 'class' => 'narrow-form']
]); ?>

<div class="form-group">
<?=
$form->field($documentoMedicoModel, 'nombre')->textInput([
    'maxlength' => true,
    'placeholder' => 'Ingrese el nombre del
documento'
])->label('Nombre del documento') ?>
</div>

<div class="form-group">
<?=
$form->field($documentoMedicoModel,
'observacion')->widget(FroalaEditorWidget::className(), [
    'clientOptions' => [
        'toolbarInline' => false,
        'theme' => 'royal',
        'language' => 'es',
        'height' => 150,
        'pluginsEnabled' => [
            'align', 'charCounter',
            'codeBeautifier', 'codeView', 'colors',
            'draggable', 'entities', 'fontFamily',
            'fontSize', 'fullscreen',
            'inlineStyle',
            'lineBreaker', 'link', 'lists',
            'paragraphFormat', 'paragraphStyle',
            'quickInsert', 'quote', 'save',
            'table', 'url', 'wordPaste'
        ]
    ]
]); ?>
```

```
        ]  
    ]) //EXTENSION QUE PERMITE QUE  
EL CAMPO DE TEXTO PUEDA MOSTRARSE COMO UN EDITOR DE  
TEXTO FUNCIONAL ?>  
    </div>  
  
    <div class="form-group">  
        <?= $form->field($documentoMedicoModel,  
        'ruta')->widget(FileInput::classname(), [  
            'options' => ['accept' => 'pdf'],  
            'pluginOptions' => [  
                'showUpload' => false,  
                'showCancel' => false,  
                'allowedFileExtensions' => ['pdf',  
                'jpg', 'xlsx'],  
  
                ],  
            ])->label('Archivo') ?>  
    </div>  
  
    <div class="form-group">  
        <?= Html::submitButton('Subir archivo <i  
        class="fa fa-upload"></i>', ['class' => 'btn btn-warning float-right']) ?>  
    </div>  
  
        <?php ActiveForm::end(); ?>  
    </div>  
    </div>  
    </div>  
    <div class="row justify-content-center">  
  
        <div class="col-md-12 mt-3">  
            <div class="card">  
                <div class="card-body">  
                    <?php  
                    $searchModel = new  
DocumentoMedicoSearch();  
                    $params =  
                    Yii::$app->request->queryParams;  
  
                    $params[$searchModel->formName()]['empleado_id'] = $model->id;
```

```
$dataProvider =  
$searchModel->search($params);  
?>  
  
<?php Pjax::begin(); ?>  
  
<?= GridView::widget([  
    'dataProvider' => $dataProvider,  
    'filterModel' => $searchModel,  
    'columns' => [  
        ['class' => 'yii\grid\SerialColumn'],  
        [  
            'attribute' => 'nombre',  
            'value' => 'nombre',  
            'filter' => false,  
            'options' => ['style' => 'width:  
20%;'],  
        ],  
        [  
            'attribute' => 'fecha_subida',  
            'format' => 'raw',  
            'value' => function ($model) {  
                setlocale(LC_TIME,  
"es_419.UTF-8");  
                return strftime('%A, %d de %B  
de %Y', strtotime($model->fecha_subida));  
            },  
            'filter' => DatePicker::widget([  
                'model' => $searchModel,  
                'attribute' => 'fecha_subida',  
                'language' => 'es',  
                'dateFormat' => 'php:Y-m-d',  
                'options' => [  
                    'class' => 'form-control',  
                    'autocomplete' => 'off',  
                ],  
                'clientOptions' => [  
                    'changeYear' => true,  
                    'changeMonth' => true,  
                    'yearRange' => '-100:+0',  
                ],  
            ],  
        ],  
    ]  
]);
```

```
        ]),
    ],
    [
        'attribute' => 'observacion',
        'format' => 'html',
        'value' => function ($model) {
            return
                \yii\helpers\Html::decode($model->observacion);
        },
        'filter' => false,
        'options' => ['style' => 'width:
51%;'],
    ],
    [
        //ACCIONES QUE SE PUEDEN
        REALIZAR CON LOS REGISTROS DE LOS DOCUMENTOS
        MEDICOS
            // SE VERIFICA LOS TIPOS DE
            ARCHIVO QUE PUEDE SER VISIBLES DESDE EL NAVEGADOR Y
            LOS
                //QUE NO, SOLO SE PERMITE
                QUE SEAN DESCARGADOS
                    'class' =>
                    'hail812\adminlte3\yii\grid\ActionColumn',
                        //template' => '{view} {delete}
{download}',
                            'template' =>
                            Yii::$app->user->can('acciones-documentos-medicos') ? '{view}
{delete} {download}' : '{view} {download}',

                    'buttons' => [
                        'view' => function ($url,
$model) {
                            // Verificar si el archivo es
                            PDF o imagen
                                $extensionsToShowEye =
                                ['pdf', 'jpg', 'jpeg', 'png', 'gif'];
                                    $fileExtension =
pathinfo($model->ruta, PATHINFO_EXTENSION);

                            if
                            (in_array(strtolower($fileExtension), $extensionsToShowEye)) {

```

```
return Html::a('<i  
class="far fa-eye"></i>', ['documento-medico/open', 'id' => $model->id],  
[  
    'target' => '_blank',  
    'title' => 'Ver archivo',  
    'class' => 'btn btn-info  
btn-xs',  
    'data-pjax' => "0"  
]);  
} else {  
    return "  
"}  
},  
'delete' => function ($url,  
$model) {  
    return Html::a('<i class="fas  
fa-trash"></i>', ['documento-medico/delete', 'id' => $model->id,  
'empleado_id' => $model->empleado_id], [  
        'title' => Yii::t('yii',  
'Eliminar'),  
        'data-confirm' => Yii::t('yii',  
'¿Estás seguro de que deseas eliminar este elemento?'),  
        'data-method' => 'post',  
        'class' => 'btn btn-danger  
btn-xs',  
    ]);  
},  
'download' => function ($url,  
$model) {  
    return Html::a('<i class="fas  
fa-download"></i>', ['documento-medico/download', 'id' => $model->id], [  
        'title' => 'Descargar  
archivo',  
        'class' => 'btn btn-success  
btn-xs',  
        'data-pjax' => "0"  
    ]);  
},  
],  
],
```

```
],
'summaryOptions' => ['class' =>
'summary mb-2'],
'pager' => [
'class' => 'yii\bootstrap4\LinkPager',
],
'tableOptions' => ['class' =>
'no-style-gridview'],
'rowOptions' => function ($model,
$key, $index, $grid) {
    return ['class' => 'no-style-gridview'];
},
]);
?>

<?php Pjax::end(); ?>
</div>
</div>
</div>
</div>
</div>
```

- antecedente\_hereditario.

Descripción. Código que tiene la funcionalidad de que se pueda crear registros sobre antecedentes hereditarios del empleado, emplea una tabla que permite seleccionar la enfermedad y de qué familiar o familiares ha sido heredada, a su vez, funciona para visualizar este registro como tal. Dependiendo de los permisos es posible editar y agregar esta información o simplemente visualizar.

```
<?php
//IMPORTACIONES
use yii\helpers\Html;
use kartik\form\ActiveForm;
```

```
$antecedentesExistentes = [];
$observacionGeneral = "";
$descripcionAntecedentes = "";
```

```
//SE ASIGNA EL PERMISO PARA PODER EDITAR SOBRE LOS
REGISTROS A UNA VARIABLE
```

```
$editable = Yii::$app->user->can('editar-expediente-medico');

//SE OBTIENEN LOS REGISTROS EXISTENTES
if ($antecedentes) {
    foreach ($antecedentes as $antecedente) {

        $antecedentesExistentes[$antecedente->cat_antecedente_hereditario
_id][$antecedente->parentezco] = true;
        if (empty($observacionGeneral)) {
            $observacionGeneral = $antecedente->observacion;
        }
    }
}

?>

<?php
//FORMULARIO DE TABLA QUE PERMITE SELECCIONAR LA
ENFERMEDAD Y AL FAMILAR
$form = ActiveForm::begin(['action' => ['empleado/view', 'id' =>
$model->id]]); ?>
    <div class="row">
        <div class="col-md-12">
            <div class="card">
                <div class="card-header gradient-blue
text-white text-center">
                    <h2>Antecedentes Hereditarios</h2>
                </div>
                <div class="card-body bg-light">
                    <div class="row">
                        <div class="col-md-8">
                            <div class="table-responsive">
                                <table class="table
table-bordered">
                                    <thead>
                                        <tr>
                                            <th>Enfermedad</th>
                                            <th>Abuelos</th>
                                            <th>Hermanos</th>
                                            <th>Madre</th>
                                            <th>Padre</th>
                                        </tr>
                                    </thead>
```

```
</thead>
<tbody>
    <?php foreach
($catAntecedentes as $catAntecedente) : ?>
        <tr>
            <td><?=
Html::encode($catAntecedente->nombre) ?></td>
                <?php foreach (['Abuelos',
'Hermanos', 'Madre', 'Padre'] as $parentezco) : ?>
                    <td>
                        <?php
// Verifica si el
usuario tiene permiso para editar

// Si es editable,
muestra el checkbox para editar
// Si no es editable,
muestra solo el estado actual sin checkbox
if ($editable) {
    echo
Html::checkbox("AntecedenteHereditario[$catAntecedente->id][$parentezco]",
isset($antecedentesExistentes[$catAntecedente->id][$parentezco]), [
    'value' => 1,
    'label' => ",
]);
} else {
// Muestra solo el
estado actual sin checkbox
$checked =
isset($antecedentesExistentes[$catAntecedente->id][$parentezco]) &&
$antecedentesExistentes[$catAntecedente->id][$parentezco] == 1;
echo $checked ?
'X' : ";
}
?>
</td>
<?php endforeach; ?>
</tr>
<?php endforeach; ?>
</tbody>
</table>
```

```
</div>
</div>
<div class="col-md-4 d-flex flex-column
justify-content-between" style="height: 100%;">
    <div class="form-group text-center">
        <?= Html::label('Observaciones',
'observacion_general') ?>
        <?=
        Html::textarea('observacion_general', $observacionGeneral, [
            'class' => 'form-control',
            'rows' => 5,
            'style' => 'width: 100%;',
            'readonly' => !$editable, // Hace
que el campo sea de solo lectura si no tiene permiso
        ]) ?>
    </div>
    <br>
    <div class="form-group mt-auto d-flex
justify-content-end">
        <?php
        // Si tiene permiso, muestra el botón
de guardar
        if ($editable) {
            echo
        Html::submitButton('Guardar     <i class="fa
fa-save"></i>', ['class' => 'btn btn-success']);
        }
        ?>
    </div>
    </div>
    <div class="alert alert-white
custom-alert" role="alert">
        <i class="fa fa-exclamation-circle"
aria-hidden="true"></i> Causas de muerte, malformaciones
congénitas, diabetes, cardiopatías, hipertensión arterial, infartos,
ateroesclerosis, accidentes vasculares, neuropatías, tuberculosis,
artropatías, hemopatías, sida, sífilis, hemopatías, neoplasias,
consanguinidad, alcoholismo, toxicomanías.
    </div>
    </div>
    </div>
</div>
```

```
</div>
</div>
<?php ActiveForm::end(); ?>
```

- antecedente\_patologico.

Descripción. Código que tiene la funcionalidad de que se pueda crear registros sobre antecedentes patológicos del empleado, también funciona para visualizar este registro como tal. Dependiendo de los permisos es posible editar y agregar esta información o simplemente visualizar.

```
<?php
//IMPORTACIONES
use yii\helpers\Html;
use froala\froalaeditor\FroalaEditorWidget;
use yii\helpers\HtmlPurifier;
use kartik\form\ActiveForm;

$editable = Yii::$app->user->can('editar-expediente-medico');

?>

<?php
//FORMULARIO PARA AGREGAR O EDITAR INFORMACIÓN DE
REGISTRO
$form = ActiveForm::begin(['action' =>
['empleado/antecedente-patologico', 'id' => $model->id]]); ?>


## Antecedentes Patológicos



<?php
// Verifica si el usuario tiene permiso para
editar
$editable =
Yii::$app->user->can('editar-expediente-medico');
?>


```

```
<?php if ($editable) : ?>
<div class="form-group">
<?=
$form->field($antecedentePatologico,
'descripcion_antecedentes')->widget(FroalaEditorWidget::className(),
[
    'clientOptions' => [
        'toolbarInline' => false,
        'theme' => 'royal', // optional:
        'dark, red, gray, royal
        'language' => 'es', // optional: ar,
        'bs, cs, da, de, en_ca, en_gb, en_us ...
        'height' => 400,
        'pluginsEnabled' => [
            'align', 'charCounter',
            'codeBeautifier', 'codeView', 'colors',
            'entities', 'fontFamily',
            'fontSize', 'fullscreen',
            'inlineStyle',
            'lineBreaker', 'link', 'lists',
            'paragraphFormat', 'paragraphStyle',
            'quickInsert', 'quote', 'save',
            'table', 'url', 'wordPaste'
        ],
    ]
])->label(false) ?>
</div>

<div class="form-group text-right">
<?= Html::submitButton('Guardar
&nbsp; &nbsp; <i class="fa fa-save"></i>', ['class' => 'btn btn-success
fa-lg']) ?>
</div>
<?php else : //SI EL PERMISO ES DE
SOLO VISUALIZAR, SE MUESTRA EL PURO CONTENIDO DEL
REGISTRO?>
<div class="form-group">
<?= Html::label('Descripción de
Antecedentes Patológicos', 'descripcion_antecedentes') ?>
<?php
```

```
$contenidoHTML =
Html::decode($antecedentePatologico->descripcion_antecedentes);
echo
HtmlPurifier::process($contenidoHTML);
?>
</div>
<?php endif; ?>

<div class="alert alert-white custom-alert"
role="alert">
    <i class="fa fa-exclamation-circle"
aria-hidden="true"></i> Peso al nacer, anomalías perinatales,
desarrollo físico y mental, y el esquema básico de vacunación.
    </div>
    </div>
    </div>
    </div>
    </div>
    <?php ActiveForm::end(); ?>
```

- antecedente\_no\_patologico.

Descripción. Código que tiene la funcionalidad de que se pueda crear registros sobre antecedentes no patológicos del empleado, también funciona para visualizar este registro como tal. Dependiendo de los permisos es posible editar y agregar esta información o simplemente visualizar.

```
<?php
//IMPORTACIONES

use yii\helpers\Html;
use froala\froalaeditor\FroalaEditorWidget;
use yii\helpers\HtmlPurifier;
use kartik\form\ActiveForm;

//SE ASIGNA EL PERMISO DE EDITAR SOBRE LOS REGISTROS A
//UNA VARIABLE
$editable = Yii::$app->user->can('editar-expediente-medico');

?>
```

```
<?php
//FORMULARIO QUE SE ENCARGA DE MOSTRAR LA
INFORMACIÓN EN LOS REGITROS O QUE PERMITE EDITAR EL
REGISTRO
$form = ActiveForm::begin(['action' => ['empleado/no-patologicos', 'id'
=> $model->id]]); ?>
<div class="row">
    <div class="col-md-12">

        <div class="card">
            <div class="card-header gradient-blue text-white text-center">
                <h2>Antecedentes No Patológicos</h2>
                <?php if (Yii::$app->user->can('editar-expediente-medico')) {
?
                <div class="float-submit-btn">
                    <?= Html::submitButton('<i class="fa fa-save"></i>', [
'class' => 'btn btn-success']) ?>
                </div>

                <?php } ?>

            </div>
            <div class="card-body">
                <div class="container">
                    <div class="card">
                        <div class="card-header custom-nopato text-white
text-left">
                            <h5>ACTIVIDAD FISICA</h5>
                        </div>
                        <div class="card-body bg-light">
                            <div class="row">
                                <div class="col-md-8">
                                    <div class="row">
                                        <div class="col-6 col-sm-4">
                                            <div class="custom-control
custom-checkbox">
                                                <?=
                                                Html::checkbox('AntecedenteNoPatologico[p_ejercicio]', [
$antecedenteNoPatologico->p_ejercicio, [
'class' => 'custom-control-input',
'id' => 'p_ejercicio',
]
)
                                            </div>
                                        </div>
                                    </div>
                                </div>
                            </div>
                        </div>
                    </div>
                </div>
            </div>
        </div>
    </div>
</div>
```

```
'disabled' => !$editable // Deshabilitar
si no tiene permiso
]) ?>
<?= Html::label('¿Realiza ejercicio?',
'p_ejercicio', ['class' => 'custom-control-label']) ?>
</div>
</div>
<div class="col-6 col-sm-6"
id="ejercicio-minutos-container">
<?= Html::label('Minutos al día',
'p_minutos_x_dia_ejercicio') ?>
<?= Html::input('number',
'AntecedenteNoPatologico[p_minutos_x_dia_ejercicio]',
$antecedenteNoPatologico->p_minutos_x_dia_ejercicio, [
'class' => 'form-control',
'id' => 'p_minutos_x_dia_ejercicio',
'disabled' => !$editable // Deshabilitar si
no tiene permiso
]) ?>
</div>
<div class="w-100"></div>
<br>
<div class="col-6 col-sm-4">
<div class="custom-control
custom-checkbox">
<?=
Html::checkbox('AntecedenteNoPatologico[p_deporte]',
$antecedenteNoPatologico->p_deporte, [
'class' => 'custom-control-input',
'id' => 'p_deporte',
'disabled' => !$editable // Deshabilitar
si no tiene permiso
]) ?>
<?= Html::label('¿Realiza algún
deporte?', 'p_deporte', ['class' => 'custom-control-label']) ?>
</div>
</div>
<div class="col-6 col-sm-4"
id="deporte-cual-container">
<?= Html::label('¿Cuál deporte?',
'p_a_deporte') ?>
```

```
<?=
Html::textInput('AntecedenteNoPatologico[p_a_deporte]',
$antecedenteNoPatologico->p_a_deporte, [
    'class' => 'form-control',
    'id' => 'p_a_deporte',
    'disabled' => !$editable // Deshabilitar si
no tiene permiso
]) ?>
</div>
<div class="col-6 col-sm-4"
id="deporte-frecuencia-container">
    <?= Html::label('Frecuencia con la que
practica', 'p_frecuencia_deporte') ?>
    <?=
Html::textInput('AntecedenteNoPatologico[p_frecuencia_deporte]',
$antecedenteNoPatologico->p_frecuencia_deporte, [
    'class' => 'form-control',
    'id' => 'p_frecuencia_deporte',
    'disabled' => !$editable // Deshabilitar si
no tiene permiso
]) ?>
</div>
<div class="w-100"></div>
<br>
<div class="col-6 col-sm-4">
    <?= Html::label('Horas que duerme por
día', 'p_horas_sueño') ?>
    <?= Html::input('number',
'AntecedenteNoPatologico[p_horas_sueño]',
$antecedenteNoPatologico->p_horas_sueño, [
    'class' => 'form-control',
    'id' => 'p_horas_sueño',
    'disabled' => !$editable // Deshabilitar si
no tiene permiso
]) ?>
</div>
<div class="col-6 col-sm-6">
    <div class="custom-control
custom-checkbox">
        <?=
Html::checkbox('AntecedenteNoPatologico[p_dormir_dia]',
$antecedenteNoPatologico->p_dormir_dia, [
```

```
'class' => 'custom-control-input',
'id' => 'p_dormir_dia',
'disabled' => !$editable // Deshabilitar
si no tiene permiso
]) ?>
<?= Html::label('¿Duerme durante el
día?', 'p_dormir_dia', ['class' => 'custom-control-label']) ?>
</div>
</div>
</div>
</div>

<div class="w-100"></div>
<br>

<div class="form-group">
<?= Html::label('Observaciones',
'observacion_actividad_fisica') ?>
<?php
if ($editable) {
    echo $form->field($antecedenteNoPatologico,
'observacion_actividad_fisica')->widget(FroalaEditorWidget::className(),
[
    'options' => [
        'id' => 'content'
    ],
    'clientOptions' => [
        'toolbarInline' => false,
        'theme' => 'royal', // optional: dark, red,
gray, royal
        'language' => 'es', // optional: ar, bs, cs,
da, de, en_ca, en_gb, en_us ...
    'height' => 150,
    'pluginsEnabled' => [
        'align',
        'charCounter',
        'codeBeautifier',
        'codeView',
        'colors',
        'draggable',
        'emoticons',
        'entities',

```

```
'fontFamily',
'fontSize',
'fullscreen',
'inlineStyle',
'lineBreaker',
'link',
'lists',
'paragraphFormat',
'paragraphStyle',
'quickInsert',
'quote',
'save',
'table',
'url',
'wordPaste'
]
]
])
])->label(false);
} else {
    // Si no tiene permiso, mostrar el texto plano
    $htmlcont =
        Html::decode($antecedenteNoPatologico->observacion_actividad_fisica);
        echo HtmlPurifier::process($htmlcont);
    }
?>
</div>

</div>
</div>
</div>
<div class="alert alert-white custom-alert" role="alert">
    <i class="fa fa-exclamation-circle" aria-hidden="true"></i> Peso al nacer, anomalías perinatales, desarrollo físico y mental, y el esquema básico de vacunación.
</div>

<?php
///SCRIPT QUE SE ENCARGA DE MOSTRAR CAMPOS OCULTOS
//PERMITE QUE SE MUESTREN ESOS CAMPOS EN CASO DE QUE
SEAN VERDAEROS
```

### //EN RELACIÓN A OTRO CAMPO

```
$script = <<< JS
$(document).ready(function() {
    function toggleEjercicioFields() {
        if ($('#p_ejercicio').is(':checked')) {
            $('#ejercicio-minutos-container').show();
        } else {
            $('#ejercicio-minutos-container').hide();
        }
    }

    function toggleDeporteFields() {
        if ($('#p_deporte').is(':checked')) {
            $('#deporte-cual-container').show();
            $('#deporte-frecuencia-container').show();
        } else {
            $('#deporte-cual-container').hide();
            $('#deporte-frecuencia-container').hide();
        }
    }

    toggleEjercicioFields();
    toggleDeporteFields();

    $('#p_ejercicio').change(function() {
        toggleEjercicioFields();
    });

    $('#p_deporte').change(function() {
        toggleDeporteFields();
    });
});

JS;
$this->registerJs($script);
?>
```

<?php //SE SEPARAN LOS CAMPOS EN DIFERENTES TARJETAS,  
SIN EMBARGO, SIGUEN SIENDO PARTE DEL MISMO  
FORMULARIO?>

```
<div class="card">
```

```
<div class="card-header custom-nopato text-white text-left">
    <h5>HABITOS ALIMENTICIOS</h5>
</div>
<div class="card-body bg-light">
    <div class="row">
        <!-- Columna izquierda con los campos -->
        <div class="col-md-8">
            <div class="row">
                <div class="col-6 col-sm-3">
                    <div class="custom-control custom-checkbox">
                        <?= Html::checkbox('AntecedenteNoPatologico[p_desayuno]', $antecedenteNoPatologico->p_desayuno, [
                            'class' => 'custom-control-input',
                            'id' => 'p_desayuno',
                            'disabled' => !$editable // Desabilitar si no tiene permiso
                        ]) ?>
                        <?= Html::label('¿Desayuna?', 'p_desayuno', ['class' => 'custom-control-label']) ?>
                    </div>
                </div>
                <div class="col-6 col-sm-6">
                    <?= Html::label('Número de comidas al día', 'p_comidas_x_dia') ?>
                    <?= Html::input('number', 'AntecedenteNoPatologico[p_comidas_x_dia]', $antecedenteNoPatologico->p_comidas_x_dia, [
                        'class' => 'form-control',
                        'disabled' => !$editable // Desabilitar si no tiene permiso
                    ]) ?>
                </div>
            </div>
        </div>
    </div>
</div>
```

```
</div>

<div class="w-100"></div>

<br>
<div class="col-6 col-sm-3">

    <div class="custom-control
custom-checkbox">
        <?=
Html::checkbox('AntecedenteNoPatologico[p_cafe]',
$antecedenteNoPatologico->p_cafe, [
    'class' => 'custom-control-input',
    'id' => 'p_cafe',
    'disabled' => !$editable // Deshabilitar
si no tiene permiso

]) ?>
<?= Html::label('¿Toma café?', 'p_cafe',
['class' => 'custom-control-label']) ?>
</div>

</div>

<div class="col-6 col-sm-6"
id="cafe-x-dia-container">
    <?= Html::label('Tazas de café al día',
'p_tazas_x_dia') ?>
        <?= Html::input('number',
'AntecedenteNoPatologico[p_tazas_x_dia]',
$antecedenteNoPatologico->p_tazas_x_dia, ['class' => 'form-control',
'disabled' => !$editable]) ?>
    </div>

<div class="w-100"></div>

<br>
<div class="col-6 col-sm-6">
```

```
<div class="custom-control
custom-checkbox">
    <?=
    Html::checkbox('AntecedenteNoPatologico[p_refresco]',
    $antecedenteNoPatologico->p_refresco, [
        'class' => 'custom-control-input',
        'id' => 'p_refresco',
        'disabled' => !$editable
    ]) ?>
    <?= Html::label('¿Toma refresco?', 'p_refresco', ['class' => 'custom-control-label']) ?>
</div>
</div>

<div class="w-100"></div>
<br>
<div class="col-6 col-sm-3">
    <div class="custom-control
custom-checkbox">
        <?=
        Html::checkbox('AntecedenteNoPatologico[p_dieta]',
        $antecedenteNoPatologico->p_dieta, [
            'class' => 'custom-control-input',
            'id' => 'p_dieta',
            'disabled' => !$editable
        ]) ?>
        <?= Html::label('¿Sigue alguna dieta?', 'p_dieta', ['class' => 'custom-control-label']) ?>
    </div>
    </div>
    <div class="col-6 col-sm-6"
id="info-dieta-container">
        <div class="form-group">
            <?= Html::label('Información sobre la
dieta', 'p_info_dieta') ?>
            <?=
            Html::textarea('AntecedenteNoPatologico[p_info_dieta]',
            $antecedenteNoPatologico->p_info_dieta, ['class' => 'form-control',
            'rows' => 6, 'disabled' => !$editable]) ?>
        </div>
    </div>
</div>
```

```
</div>

</div>
</div>
<!-- Columna derecha con el textarea -->

<div class="w-100"></div>

<br>

<div class="form-group">
    <?= Html::label('Observaciones',
'observacion_comida') ?>

    <?php
        if ($editable) {
            echo $form->field($antecedenteNoPatologico,
'observacion_comida')->widget(FroalaEditorWidget::className(), [
                'options' => [
                    'id' => 'content'
                ],
                ])-
            ])->label(false);
        } else {
            // Si no tiene permiso, mostrar el texto plano
            $htmlcont =
                Html::decode($antecedenteNoPatologico->observacion_comida);
            echo HtmlPurifier::process($htmlcont);
        }
    ?>
</div>

</div>
</div>
</div>

<?php
```

```
///SCRIPT QUE SE ENCARGA DE MOSTRAR CAMPOS  
OCULTOS  
//PERMITE QUE SE MUESTREN ESOS CAMPOS EN CASO DE QUE  
SEAN VERDAEROS  
//EN RELACIÓN A OTRO CAMPO  
$script = <<< JS  
$(document).ready(function() {  
    function toggleCafeFields() {  
        if ($('#p_cafe').is(':checked')) {  
            $('#cafe-x-dia-container').show();  
        } else {  
            $('#cafe-x-dia-container').hide();  
        }  
    }  
  
    function toggleDietaFields() {  
        if ($('#p_dieta').is(':checked')) {  
            $('#info-dieta-container').show();  
        } else {  
            $('#info-dieta-container').hide();  
        }  
    }  
  
    toggleCafeFields();  
    toggleDietaFields();  
    $('#p_cafe').change(function() {  
        toggleCafeFields();  
    });  
  
    $('#p_dieta').change(function() {  
        toggleDietaFields();  
    });  
});  
JS;  
$this->registerJs($script);  
?>
```

```
<div class="card">
```

```
<div class="card-header custom-nopato text-white text-left">
    <h5>ALCOHOLISMO</h5>
</div>
<div class="card-body bg-light">
    <div class="row">
        <!-- Columna izquierda con los campos -->
        <div class="col-md-7">
            <div class="row">
                <div class="col-6 col-sm-4">
                    <div class="custom-control custom-checkbox">
                        <?= Html::checkbox('AntecedenteNoPatologico[p_alcohol]', $antecedenteNoPatologico->p_alcohol, [
                            'class' => 'custom-control-input',
                            'id' => 'p_alcohol',
                            'disabled' => !$editable
                        ]) ?>
                        <?= Html::label('¿Consume alcohol?', 'p_alcohol', ['class' => 'custom-control-label']) ?>
                    </div>
                </div>
                <div class="col-6 col-sm-6" id="consumo-alcohol-container">
                    <!-- Campo dropdown -->
                    <div class="form-group">
                        <?= Html::label('Frecuencia de Consumo de Alcohol', 'p_frecuencia_alcohol') ?>
                        <?= Html::dropDownList('AntecedenteNoPatologico[p_frecuencia_alcohol]', $antecedenteNoPatologico->p_frecuencia_alcohol, [
                            'Casual' => 'Casual',
                            'Moderado' => 'Moderado',
                            'Intenso' => 'Intenso',
                        ], [
                            'class' => 'form-control',
                            'prompt' => 'Seleccione la frecuencia',
                            'disabled' => !$editable
                        ]) ?>
                    </div>
                </div>
            </div>
        </div>
    </div>
</div>
```

```
<?= Html::label('Edad a la que comenzó a  
béber', 'p_edad_alcoholismo') ?>
```

```
<?= Html::input('number',  
'AntecedenteNoPatologico[p_edad_alcoholismo]',  
$antecedenteNoPatologico->p_edad_alcoholismo, ['class' =>  
'form-control', 'disabled' => !$editable]) ?>
```

```
<?= Html::label('Copas de licor/vino al día',  
'p_copas_x_dia') ?>
```

```
<?= Html::input('number',  
'AntecedenteNoPatologico[p_copas_x_dia]',  
$antecedenteNoPatologico->p_copas_x_dia, ['class' => 'form-control',  
'disabled' => !$editable]) ?>
```

```
<?= Html::label('Número de cervezas al  
día', 'p_cervezas_x_dia') ?>
```

```
<?= Html::input('number',  
'AntecedenteNoPatologico[p_cervezas_x_dia]',  
$antecedenteNoPatologico->p_cervezas_x_dia, ['class' =>  
'form-control', 'disabled' => !$editable]) ?>  
</div>
```

```
<div class="w-100"></div>  
<br>
```

```
<div class="col-6 col-sm-4">  
<div class="custom-control  
custom-checkbox">
```

```
<?=  
Html::checkbox('AntecedenteNoPatologico[p_ex_alcoholico]',  
$antecedenteNoPatologico->p_ex_alcoholico, [  
    'class' => 'custom-control-input',  
    'id' => 'p_ex_alcoholico',  
    'disabled' => !$editable  
) ?>  
<?= Html::label('Ex-alcoholico',  
'p_ex_alcoholico', ['class' => 'custom-control-label']) ?>  
</div>
```

```
</div>
<div class="col-6 col-sm-6"
id="ex-alcoholico-container">
    <?= Html::label('Edad en la que dejo de
beber', 'p_edad_fin_alcoholismo') ?>
        <?= Html::input('number',
'AntecedenteNoPatologico[p_edad_fin_alcoholismo]',
$antecedenteNoPatologico->p_edad_fin_alcoholismo, ['class' =>
'form-control', 'disabled' => !$editable]) ?>
            </div>
        </div>
    </div>
    <!-- Columna derecha con el textarea -->

    <div class="w-100"></div>
    <br>

<div class="form-group">
    <?= Html::label('Observaciones',
'observacion_alcoholismo') ?>

    <?php
    if ($editable) {
        echo $form->field($antecedenteNoPatologico,
'observacion_alcoholismo')->widget(FroalaEditorWidget::className(),
[
    'options' => [
        'id' => 'content'
    ],
    ])>label(false);
    } else {
        // Si no tiene permiso, mostrar el texto plano
        $htmlcont =
        Html::decode($antecedenteNoPatologico->observacion_alcoholismo);
        echo HtmlPurifier::process($htmlcont);
    }
    ?>
</div>
```

```
</div>
</div>
</div>

<?php
///SCRIPT QUE SE ENCARGA DE MOSTRAR CAMPOS
OCULTOS
//PERMITE QUE SE MUESTREN ESOS CAMPOS EN CASO DE QUE
SEAN VERDAEROS
//EN RELACIÓN A OTRO CAMPO
$script = <<< JS
$(document).ready(function() {

    function toggleAlcoholFields() {
        if ($('#p_alcohol').is(':checked')) {
            $('#consumo-alcohol-container').show();
        } else {
            $('#consumo-alcohol-container').hide();
        }
    }

    function toggleExAlcoholicoFields() {
        if ($('#p_ex_alcoholico').is(':checked')) {
            $('#ex-alcoholico-container').show();
        } else {
            $('#ex-alcoholico-container').hide();
        }
    }

    toggleAlcoholFields();
    toggleExAlcoholicoFields();

    $('#p_alcohol').change(function() {
        toggleAlcoholFields();
    });

    $('#p_ex_alcoholico').change(function() {
        toggleExAlcoholicoFields();
    });
});
```

```
    });
  });
JS;
$this->registerJs($script);
?>

<div class="card">
  <div class="card-header custom-nopato text-white text-left">
    <h5>TABAQUISMO</h5>

  </div>
  <div class="card-body bg-light">
    <div class="row">
      <!-- Columna izquierda con los campos -->
      <div class="col-md-8">
        <div class="row">
          <div class="col-6 col-sm-4">
            <div class="custom-control
custom-checkbox">
              <?=
Html::checkbox('AntecedenteNoPatologico[p_fuma]',
$antecedenteNoPatologico->p_fuma, [
  'class' => 'custom-control-input',
  'id' => 'p_fuma',
  'disabled' => !$editable
]) ?>
              <?= Html::label('¿Fúma?', 'p_fuma',
['class' => 'custom-control-label']) ?>
            </div>
          </div>
          <div class="col-6 col-sm-6"
id="tabaquismo-container">
            <!-- Campo dropdown -->
            <div class="form-group">
              <?= Html::label('Frecuencia de Consumo
de Tabaco', 'p_frecuencia_tabaquismo') ?>
              <?=
Html::dropDownList('AntecedenteNoPatologico[p_frecuencia_tabaquis
mo]', $antecedenteNoPatologico->p_frecuencia_tabaquismo, [
```

```
'Casual' => 'Casual',
'Moderado' => 'Moderado',
'Intenso' => 'Intenso',
], [
    'class' => 'form-control',
    'prompt' => 'Seleccione la frecuencia',
    'disabled' => !$editable
]) ?>
</div>
<?= Html::label('Edad a la que comenzó a
fumar', 'p_edad_tabaquismo') ?>
    <?= Html::input('number',
'AntecedenteNoPatologico[p_edad_tabaquismo]',
$antecedenteNoPatologico->p_edad_tabaquismo, ['class' =>
'form-control', 'disabled' => !$editable]) ?>

    <?= Html::label('Número de cigarros al día',
'p_no_cigarros_x_dia') ?>
        <?= Html::input('number',
'AntecedenteNoPatologico[p_no_cigarros_x_dia]',
$antecedenteNoPatologico->p_no_cigarros_x_dia, ['class' =>
'form-control', 'disabled' => !$editable]) ?>

    </div>
    <div class="w-100"></div>
    <br>
    <div class="col-6 col-sm-4">
        <div class="custom-control
custom-checkbox">
            <?=
                Html::checkbox('AntecedenteNoPatologico[p_ex_fumador]',
$antecedenteNoPatologico->p_ex_fumador, [
                    'class' => 'custom-control-input',
                    'id' => 'p_ex_fumador',
                    'disabled' => !$editable
]) ?>
            <?= Html::label('Ex-Fumador',
'p_ex_fumador', ['class' => 'custom-control-label']) ?>
        </div>
    </div>
```

```
<div class="col-6 col-sm-6"
id="ex-fumador-container">
    <?= Html::label('Edad en la que dejo de
fumar', 'p_edad_fin_tabaquismo') ?>
    <?= Html::input('number',
'AntecedenteNoPatologico[p_edad_fin_tabaquismo]',
$antecedenteNoPatologico->p_edad_fin_tabaquismo, ['class' =>
'form-control', 'disabled' => !$editable]) ?>
</div>
<div class="w-100"></div>
<br>
<div class="col-6 col-sm-4">
    <div class="custom-control
custom-checkbox">
        <?=
        Html::checkbox('AntecedenteNoPatologico[p_fumador_pasivo]',
$antecedenteNoPatologico->p_fumador_pasivo, [
            'class' => 'custom-control-input',
            'id' => 'p_fumador_pasivo',
            'disabled' => !$editable
        ]) ?>
        <?= Html::label('Fumador Pasivo',
'p_fumador_pasivo', ['class' => 'custom-control-label']) ?>
    </div>
    </div>

    </div>
</div>

<div class="w-100"></div>
<br>
<div class="form-group">
    <?= Html::label('Observaciones',
'observacion_tabaquismo') ?>

    <?php
    if ($editable) {
        echo $form->field($antecedenteNoPatologico,
'observacion_tabaquismo')->widget(FroalaEditorWidget::className(), [
            'options' => [
                'id' => 'content'
            ],
        ],
    },

```

```
        ])>label(false);
    } else {
        // Si no tiene permiso, mostrar el texto plano
        $htmlcont =
Html::decode($antecedenteNoPatologico->observacion_tabaquismo);
        echo HtmlPurifier::process($htmlcont);
    }
    ?>
</div>

</div>
</div>
</div>

<?php
///SCRIPT QUE SE ENCARGA DE MOSTRAR CAMPOS
OCULTOS
//PERMITE QUE SE MUESTREN ESOS CAMPOS EN CASO DE QUE
SEAN VERDAEROS
//EN RELACIÓN A OTRO CAMPO
$script = <<< JS
$(document).ready(function() {
    function toggleTabaquismoFields() {
        if ($('#p_fuma').is(':checked')) {
            $('#tabaquismo-container').show();
        } else {
            $('#tabaquismo-container').hide();
        }
    }

    function toggleExFumadorFields() {
        if ($('#p_ex_fumador').is(':checked')) {
            $('#ex-fumador-container').show();
        } else {
            $('#ex-fumador-container').hide();
        }
    }
})
```

```
toggleTabaquismoFields();
toggleExFumadorFields();

$('#p_fuma').change(function() {
    toggleTabaquismoFields();
});

$('#p_ex_fumador').change(function() {
    toggleExFumadorFields();
});
});

JS;
$this->registerJs($script);
?>

<div class="card">
    <div class="card-header custom-nopato text-white text-left">
        <h5>OTROS</h5>

    </div>
    <div class="card-body bg-light">
        <div class="row">
            <!-- Columna izquierda con los campos -->
            <div class="col-md-6">
                <div class="row">
                    <div class="form-group">
                        <?= Html::label('Religión a la que
pertenece', 'religion') ?>
                        <?=
                        Html::dropDownList('AntecedenteNoPatologico[religion]',
$antecedenteNoPatologico->religion, [
                            'Ninguna' => 'Ninguna',
                            'Católica' => 'Católica',
                            'Cristiana Evangélica' => 'Cristiana
Evangélica',
                            'Testigos de Jehová' => 'Testigos de
Jehová',
                            'Mormona' => 'Mormona',
                            'Bautista' => 'Bautista',
                            'Pentecostal' => 'Pentecostal',
                        ]
                    )
                </div>
            </div>
        </div>
    </div>
</div>
```

```
'Adventista del Séptimo Día' =>
'Adventista del Séptimo Día',
'Judía' => 'Judía',
'Budista' => 'Budista',
'Hinduista' => 'Hinduista',
'Musulmana' => 'Musulmana',
'Ateísta' => 'Ateísta',
'Agnóstica' => 'Agnóstica',
'Otra' => 'Otra'
], [
  'class' => 'form-control',
  'prompt' => 'Seleccione la religión',
  'disabled' => !$editable
]) ?>
</div>

<div class="form-group">
  <?= Html::label('¿Qué actividades realiza
en sus horas libres?', 'p_act_dias_libres') ?>
  <?=
  Html::textArea('AntecedenteNoPatologico[p_act_dias_libres]',
$antecedenteNoPatologico->p_act_dias_libres, ['class' =>
'form-control', 'rows' => 5, 'disabled' => !$editable]) ?>
  </div>
  <div class="form-group">
    <?= Html::label('¿Pasa por algunas de
estas situaciones?', 'p_situaciones') ?>
    <?=
    Html::dropDownList('AntecedenteNoPatologico[p_situaciones]',
$antecedenteNoPatologico->p_situaciones, [
  'Ninguna' => 'Ninguna',
  'Duelo' => 'Duelo',
  'Embarazos' => 'Embarazos',
  'Divorcio' => 'Divorcio',
], [
  'class' => 'form-control',
  'prompt' => 'Seleccione una opción',
  'disabled' => !$editable
]) ?>
  </div>
  <div class="form-group">
```

```
<?= Html::label('Tipo de Sangre',
'tipo_sangre') ?>
<?=
Html::dropDownList('AntecedenteNoPatologico[tipo_sangre]',
$antecedenteNoPatologico->tipo_sangre, [
'A+' => 'A+',  

'B+' => 'B+',  

'O+' => 'O+',  

'A-' => 'A-',  

'B-' => 'B-',  

'O-' => 'O-',  

'AB+' => 'AB+',  

'AB-' => 'AB-',
], [  

'class' => 'form-control',  

'prompt' => 'Seleccione el tipo de  

sangre',
'disabled' => !$editable
]) ?>
</div>
```

```
</div>
</div>
<!-- Columna derecha con el textarea -->
```

```
<div class="form-group">
<?= Html::label('Descripción de su vivienda  

(Tiene mascotas, Recursos del hogar, Etc.)', 'datos_vivienda') ?>

<?php
if ($editable) {
    echo $form->field($antecedenteNoPatologico,
'datos_vivienda')->widget(FroalaEditorWidget::className(), [
'options' => [
'id' => 'content'
],
]
```

```
        ])>label(false);
    } else {

        $htmlcont =
Html::decode($antecedenteNoPatologico->datos_vivienda);
echo HtmlPurifier::process($htmlcont);
}

?>
</div>

</div>
</div>
</div>

<div class="card">
<div class="card-header custom-nopato text-white
text-left">
<h5>CONSUMO DE DROGAS</h5>

</div>
<div class="card-body bg-light">
<div class="row">
<!-- Columna izquierda con los campos -->
<div class="col-md-8">
<div class="row">
<div class="col-6 col-sm-4">
<div class="custom-control
custom-checkbox">
<?=
Html::checkbox('AntecedenteNoPatologico[p_drogas]',
$antecedenteNoPatologico->p_drogas, [
    'class' => 'custom-control-input',
    'id' => 'p_drogas',
    'disabled' => !$editable
]) ?>
<?= Html::label('¿Consumo algún tipo de
droga?', 'p_drogas', ['class' => 'custom-control-label']) ?>
</div>
```

</div>

```
<div class="col-6 col-sm-6"
id="droga-container">

    <div class="form-group">
        <?= Html::label('Frecuencia de su
consumo', 'p_frecuencia_droga') ?>
        <?=
        Html::dropDownList('AntecedenteNoPatologico[p_frecuencia_droga]',
$antecedenteNoPatologico->p_frecuencia_droga, [
            'Casual' => 'Casual',
            'Moderado' => 'Moderado',
            'Intenso' => 'Intenso',
        ], [
            'class' => 'form-control',
            'prompt' => 'Seleccione la frecuencia',
            'disabled' => !$editable
        ]) ?>
    </div>
    <?= Html::label('¿A qué edad se inicio el
consumo?', 'p_edad_droga') ?>
    <?= Html::input('number',
'AntecedenteNoPatologico[p_edad_droga]',
$antecedenteNoPatologico->p_edad_droga, ['class' => 'form-control',
'disabled' => !$editable]) ?>
```

```
<div class="custom-control
custom-checkbox">
    <?=
    Html::checkbox('AntecedenteNoPatologico[p_droga_intravenosa]',
$antecedenteNoPatologico->p_droga_intravenosa, [
        'class' => 'custom-control-input',
        'id' => 'p_droga_intravenosa',
        'disabled' => !$editable
    ]) ?>
    <br>
```

```
<?= Html::label('¿Usa drogas
intravenosa?', 'p_droga_intravenosa', ['class' => 'custom-control-label'])
?>
</div>

</div>

<div class="w-100"></div>
<br>
<div class="col-6 col-sm-4">
    <div class="custom-control
custom-checkbox">
        <?=
        Html::checkbox('AntecedenteNoPatologico[p_ex_adicto]',
        $antecedenteNoPatologico->p_ex_adicto, [
            'class' => 'custom-control-input',
            'id' => 'p_ex_adicto',
            'disabled' => !$editable
        ]) ?>
        <?= Html::label('Ex-Adicto',
        'p_ex_adicto', ['class' => 'custom-control-label']) ?>
        </div>
    </div>
    <div class="col-6 col-sm-6"
id="ex-adicto-container">
        <?= Html::label('¿A qué edad dejó de
consumir?', 'p_edad_fin_droga') ?>
        <?= Html::input('number',
        'AntecedenteNoPatologico[p_edad_fin_droga]',
        $antecedenteNoPatologico->p_edad_fin_droga, ['class' =>
        'form-control', 'disabled' => !$editable]) ?>
        </div>
    </div>
</div>
```

```
<div class="w-100"></div>
<br>

<div class="form-group">
    <?= Html::label('Observaciones',
'observacion_droga') ?>

    <?php
        if ($editable) {
            echo $form->field($antecedenteNoPatologico,
'observacion_droga')->widget(FroalaEditorWidget::className(), [
                'options' => [
                    'id' => 'content'
                ],
                ])-
            ->label(false);
        } else {
            // Si no tiene permiso, mostrar el texto plano
            $htmlcont =
                Html::decode($antecedenteNoPatologico->observacion_droga);
            echo HtmlPurifier::process($htmlcont);
        }
    ?>
</div>

</div>
</div>
</div>

<?php
//SCRIPT QUE SE ENCARGA DE MOSTRAR CAMPOS
OCULTOS
//PERMITE QUE SE MUESTREN ESOS CAMPOS EN CASO DE QUE
SEAN VERDAEROS
//EN RELACIÓN A OTRO CAMPO
$script = <<< JS
$(document).ready(function() {
    function toggleDrogasFields() {
```

```
if ($('#p_drogas').is(':checked')) {
    $('#droga-container').show();
} else {
    $('#droga-container').hide();
}

function toggleExAdictoFields() {
    if ($('#p_ex_adicto').is(':checked')) {
        $('#ex-adicto-container').show();

    } else {
        $('#ex-adicto-container').hide();
    }
}

toggleDrogasFields();
toggleExAdictoFields();
$('#p_drogas').change(function() {
    toggleDrogasFields();
});

$('#p_ex_adicto').change(function() {
    toggleExAdictoFields();
});
};

JS;
$this->registerJs($script);
?>

<div class="card">
    <div class="card-header custom-nopato text-white text-left">
        <h5>OBSERVACION GENERAL / OTRAS OBSERVACIONES</h5>
    </div>
    <div class="card-body bg-light">
        <div class="row">
            <div class="form-group">
```

```
<?= Html::label('Observación General',
'observacion_general') ?>

<?php
if ($editable) {
    echo $form->field($antecedenteNoPatologico,
'observacion_general')->widget(FroalaEditorWidget::className(), [
    'clientOptions' => [
        'toolbarInline' => false,
        'theme' => 'royal', // optional: dark, red,
        gray, royal
        'language' => 'es', // optional: ar, bs, cs,
        da, de, en_ca, en_gb, en_us ...
        'height' => 300,
        'pluginsEnabled' => [
            'align',
            'charCounter',
            'codeBeautifier',
            'codeView',
            'colors',
            'draggable',
            'emoticons',
            'entities',
            'fontFamily',
            'fontSize',
            'fullscreen',
            'inlineStyle',
            'lineBreaker',
            'link',
            'lists',
            'paragraphFormat',
            'paragraphStyle',
            'quickInsert',
            'quote',
            'save',
            'table',
            'url',
            'wordPaste'
        ]
    ]
})->label(false);
} else {
```

```
// Si no tiene permiso, mostrar el texto plano
$htmlcont =
    Html::decode($antecedenteNoPatologico->observacion_general);
    echo HtmlPurifier::process($htmlcont);
}
?>
</div>
</div>
</div>
</div>
<?php if
(Yii::$app->user->can('editar-expediente-medico')) { ?>
<div class="form-group">

    <?= Html::submitButton('Guardar Todos los Cambios
&nbsp; &nbsp; <i class="fa fa-save"></i>', ['class' => 'btn btn-success']) ?>
</div>
<?php } ?>

</div>
<br>
</div>

</div>
</div>
</div>
<?php ActiveForm::end(); ?>
```

- antecedente\_ginecologico, antecedente\_obstrectico, antecedente\_perinatal, alergias, exploracion\_fisica, interrogatorio\_medico.

Descripción. El código de estas vistas tienen el mismo funcionamiento y lógica que se encuentra en antecedente\_no\_patologico, la única diferencia son los campos del formulario en base a los registros, en cuanto a lógica sigue la misma estructura. Los registros junto a su información se muestran, dependiendo de los permisos se puede únicamente visualizar, en caso contrario se puede editar o agregar información a estos.

- antecedente\_no\_patologico.

Descripción. Código que tiene la funcionalidad de que se pueda crear registros sobre antecedentes no patológicos del empleado, también funciona para visualizar este registro como tal. Dependiendo de los permisos es posible editar y agregar esta información o simplemente visualizar.

### Solicitud.

- index.

Descripción. Código que muestra la lista general de todas las solicitudes de los empleados. También contiene una funcionalidad donde se pueden filtrar los registros de esta lista.

```
<?php
//IMPORTACIONES
use yii\helpers\Html;
use yii\widgets\Pjax;
use kartik\select2\Select2;
use yii\grid\GridView;
use yii\jui\DatePicker;
use yii\web\View;
use kartik\daterange\DateRangePicker;
/* @var $this yii\web\View */
/* @var $searchModel app\models\SolicitudSearch */
/* @var $dataProvider yii\data\ActiveDataProvider */

$this->title = 'Historial de solicitudes de incidencias';
$this->params['breadcrumbs'][] = $this->title;
//SE CARGAN LOS ESTILOS
$this->registerCssFile('@web/css/site.css', ['position' => View::POS_HEAD]);

?>
<div class="container-fluid">
    <div class="row">
        <div class="col-md-12">
```

```
<div class="card">
    <div class="card-header gradient-info text-white">
        <h3><?= Html::encode($this->title) ?></h3>
    </div>
    <div class="card-body">

        <?php
//LISTA QUE MUESTRA LOS REGISTROS DE SOLICITUDES QUE HA
REALIZADO LOS EMPLEADOS
Pjax::begin(['id' => 'pjax-container']);
echo GridView::widget([
    'dataProvider' => $dataProvider,
    'filterModel' => $searchModel,

    'columns' => [
        ['class' => 'yii\grid\SerialColumn'],
        [
            'class' => 'yii\grid\DataColumn',
            'header' => '',
            'format' => 'raw',
            'value' => function ($model) { //AYUDA A IDENTIFICAR EL
ESTATUS DE LA SOLICITUD Y VERIFICAR SI SON NUEVAS O YA HAN SIDO
VISUALIZADAS
                if ($model->status === 'Nueva') {
                    return '<i class="fa fa-fire" aria-hidden="true" style="color:
#ea4242 "></i>';
                } elseif ($model->status === 'Visto') {
                    return '<i class="fa fa-check-double" aria-hidden="true"
style="color: #4678fc"></i>';
                } else {
                    return '<i class="fa fa-question" aria-hidden="true"
style="color: #6c757d"></i>';
                }
            },
            'contentOptions' => ['class' => 'text-center'],
            'headerOptions' => ['class' => 'text-center'],
        ],
        [
            'attribute' => 'empleado_id',
            'label' => 'Empleado',
        ]
    ]
])
```

```
'value' => function ($model) {
    return $model->empleado ? $model->empleado->nombre . '' .
$model->empleado->apellido : 'N/A';
},
'filter' => Select2::widget([
    'model' => $searchModel,
    'attribute' => 'empleado_id',
    'data' =>
yii\helpers\ArrayHelper::map(\app\models\Empleado::find()->all(), 'id', function
($model) {
    return $model->nombre . '' . $model->apellido;
}),
'options' => ['placeholder' => 'Seleccione un empleado'],
'pluginOptions' => [
    'allowClear' => true
],
'theme' => Select2::THEME_KRAJEE_BS3,
]),
],
[
    'attribute' => 'fecha_creacion',
    'format' => 'raw',
    'value' => function ($model) {
        setlocale(LC_TIME, "es_419.UTF-8");
        return strftime('%A, %d de %B de %Y',
strtotime($model->fecha_creacion)));
    },
    'filter' => DatePicker::widget([
        'model' => $searchModel,
        'attribute' => 'fecha_creacion',
        'language' => 'es',
        'dateFormat' => 'php:Y-m-d',
        'options' => [
            'class' => 'form-control',
            'autocomplete' => 'off',
        ],
        'clientOptions' => [
            'changeYear' => true,
            'changeMonth' => true,
            'yearRange' => '-100:+0',
        ],
    ],
]
```

```
]),  
],  
  
[  
    'attribute' => 'nombre_formato',  
    'label' => 'Tipo de solicitud',  
    'value' => function ($model) {  
        return $model->nombre_formato;  
    },  
    'filter' => Select2::widget([  
        'model' => $searchModel,  
        'attribute' => 'nombre_formato',  
        'data' =>  
            \yii\helpers\ArrayHelper::map(\app\models\Solicitud::find()->select(['nombre_formato'])->distinct()->all(), 'nombre_formato', 'nombre_formato'),  
        'options' => ['placeholder' => 'Seleccione un tipo de solicitud'],  
        'pluginOptions' => [  
            'allowClear' => true  
  
        ],  
        'theme' => Select2::THEME_KRAJEE_BS3,  
    ]),  
],  
  
[  
    'class' => 'hail812\adminlte3\yii\grid\ActionColumn',  
  
    'template' => '{view}',  
  
    'buttons' => [  
  
        'view' => function ($url, $model) {  
            return Html::a('<i class="far fa-eye"></i>', $url, [  
                'title' => 'Ver solicitud',  
                'class' => 'btn btn-outline-info btn-sm',  
                'data-pjax' => "0"  
            ]);  
        },  
    ],  
];
```

```
        ],
        ],
        'summaryOptions' => ['class' => 'summary mb-2'],
        'pager' => [
            'class' => 'yii\\bootstrap4\\LinkPager',
        ],
    ]);

Pjax::end();

?>

</div>
<!--.card-body-->
</div>
<!--.card-->
</div>
<!--.col-md-12-->
</div>
<!--.row-->
</div>
```

- view.

Descripción. Código que muestra la información detallada de solicitud realizada por empleado. Esta vista muestra la información que se encuentra en el registro de la solicitud, también muestra la información del formato asociado a la solicitud. En este código se encuentra un switch que identifica el tipo de solicitud que es y mostrar la información adecuada respecto al formato.

```
<?php
//IMPORTACIONES
use yii\helpers\Html;
```

```
use yii\widgets\DetailView;
use yii\bootstrap5\Alert;
use yii\web\View;
use yii\grid\GridView;
/* @var $this yii\web\View */
/* @var $model app\models\Solicitud */
$this->registerCssFile('@web/css/site.css', ['position' =>
View::POS_HEAD]);

$this->title = $model->id;

if ($empleadoid !== null && Yii::$app->user->can('medico') ||
Yii::$app->user->can('gestor-rh')) {
//PERMISO
    $this->params['breadcrumbs'][] = ['label' => 'Empleado: ' .
$model->empleado->nombre . ' ' . $model->empleado->apellido, 'url' =>
['empleado/view', 'id' => $empleadoid]];
}
else if (Yii::$app->user->can('empleado')) {
    $this->params['breadcrumbs'][] = ['label' => 'Inicio', 'url' =>
['site/portalempleado']];
}

if (Yii::$app->user->can('gestor-rh')) {
    $this->params['breadcrumbs'][] = ['label' => 'Solicitudes', 'url' =>
['solicitud/index']];
}

$this->params['breadcrumbs'][] = ['label' => 'Solicitud ' . $model->id];

\yii\web\YiiAsset::register($this);
?>
<div class="container-fluid">
    <div class="row justify-content-center">
        <div class="col-md-8">
            <div class="card">
                <div class="card-header bg-info text-white">
                    <div class="d-flex justify-content-between">
                        <?php if (Yii::$app->user->can('medico')) { ?>
```

```
<h3> CITA MEDICA </h3>
<p> Empleado: <?= $model->empleado->nombre.'$model->empleado->apellido ?></p>

<?php }else { ?>
    <h3> Solicitud </h3>
    <p> Empleado: <?= $model->empleado->nombre.'$model->empleado->apellido ?></p>

<?php } ?>

<?php if (Yii::$app->user->can('crear-consulta-medica')) : ?>

<?php if ($model->empleado->expedienteMedico): ?>
    <?= Html::a('Nueva consulta <i class="fa fa-user-md" ></i>', ['consulta-medica/create', 'expediente_medico_id' => $model->empleado->expedienteMedico->id], [
        'class' => 'btn btn-dark mt-3 float-right fa-lg'
    ]) ?>

<?php endif; ?>

<?php endif; ?>

</div>
</div>
<div class="card-body">
    <div class="row">
        <div class="col-md-12">
            <?php foreach (Yii::$app->session->getAllFlashes() as $type => $message) { //ALERTA
                echo Alert::widget([
                    'options' => ['class' => 'alert-' . $type],
                    'body' => $message,
                ]);
            } ?>

<?php
```

```
echo DetailView::widget([
    ///MUESTRA LA INFORMACION DEL REGISTRO DE SOLICITUD
    'model' => $model,
    'attributes' => [
        [
            'label' => 'Número de empleado',
            'value' => function ($model) {
                return $model->empleado->numero_empleado;
            },
        ],
        [
            'label' => 'Tipo de formato',
            'value' => function ($model) {
                return $model->nombre_formato;
            },
        ],
        [
            'label' => 'Nombre',
            'value' => function ($model) {
                return $model->empleado ? $model->empleado->nombre . ''
. $model->empleado->apellido : 'N/A';
            },
        ],
        [
            'attribute' => 'fecha_creacion',
            'format' => 'raw',
            'value' => function ($model) {
                setlocale(LC_TIME, "es_419.UTF-8");
                return strftime("%A, %d de %B de %Y",
strptime($model->fecha_creacion));
            },
        ],
        [
    ],
]);
if ($formato) {
    $formatoAttributes = [];
    //ESTE SWITCH SE ENCARGA DE IDENTIFICAR CUALES
    ATRIBUTOS SE VAN A MOSTRAR EN EL DETAILVIEW
```

```
//YA QUE DEPENDE DEL TIPO DE SOLICITUD QUE SE HAYA  
REALIZADO SE VAN A MOSTRAR DIFERENTES DATOS  
//SE MOSTRARA LA INFORMACION DEL FORMATO ASOCIADO  
A LA SOLICITUD  
switch ($model->nombre_formato) {  
    case 'REPORTE DE TIEMPO EXTRA':  
        $formatoAttributes = [  
            [  
                'label' => 'ID',  
                'attribute' => 'solicitud_id',  
                'value' => function ($model) {  
                    return $model->solicitud_id;  
                },  
            ],  
            [  
                'label' => 'Fecha de creación',  
                'attribute' => 'created_at',  
                'value' => function ($model) {  
                    return $model->created_at;  
                },  
            ],  
            [  
                'label' => 'Actividades',  
                'attribute' => 'actividades',  
                'value' => function ($model) {  
                    echo GridView::widget([  
                        'dataProvider' => new \yii\data\ArrayDataProvider([  
                            'allModels' => $actividades,  
                            'pagination' => false,  
                        ]),  
                        'columns' => [  
                            ['class' => 'yii\grid\SerialColumn'],  
                            'fecha',  
                            'hora_inicio',  
                            'hora_fin',  
                            'actividad',  
                            'no_horas',  
                        ],  
                    ]);  
                },  
            ],  
        ];  
        echo '<div class="row">';  
        echo '<div class="col-md-12">';  
        echo '<h3>Actividades</h3>';  
        echo Gridview::widget([  
            'dataProvider' => new \yii\data\ArrayDataProvider([  
                'allModels' => $actividades,  
                'pagination' => false,  
            ]),  
            'columns' => [  
                ['class' => 'yii\grid\SerialColumn'],  
                'fecha',  
                'hora_inicio',  
                'hora_fin',  
                'actividad',  
                'no_horas',  
            ],  
        ]);  
        echo '</div>';  
        echo '</div>';  
    }  
}
```

```
echo '</div>';
echo '</div>';
break;

case 'PERMISO FUERA DEL TRABAJO':
$formatoAttributes = [
[
    'attribute' => 'hora_salida',
    'value' => function ($formato) {
        $hora = date("g:i A", strtotime($formato->hora_salida));
        return $hora;
    },
],
[
    'attribute' => 'hora_Regreso',
    'value' => function ($formato) {
        $hora = date("g:i A",
            strtotime($formato->hora_Regreso));
        return $hora;
    },
],
[
    'label' => 'Motivo',
    'attribute' => 'motivo',
    'format' => 'html',
    'value' => function ($model) {
        return
            \yii\helpers\Html::decode($model->motivoFechaPermiso->motivo);
    },
    'filter' => false,
    'options' => ['style' => 'width: 65%;'],
],
[
    'attribute' => 'horario_fecha_a_reponer',
    'value' => function ($formato) {
        $hora = date("g:i A",
            strtotime($formato->horario_fecha_a_reponer));
        return $hora;
    },
],
]
```

```
// 'nota:ntext',
[
    'label' => 'Fecha de Permiso',
    'value' => function ($formato) {

        setlocale(LC_TIME, "es_419.UTF-8");

        $fechaPermiso =
        strtotime($formato->motivoFechaPermiso->fecha_permiso);

        $fechaFormateada = strftime('%A, %B %d, %Y',
        $fechaPermiso);

        setlocale(LC_TIME, null);

        return $fechaFormateada;
    },
],
];
break;

case 'COMISION ESPECIAL':
$formatoAttributes = [

[
    'label' => 'Motivo',
    'attribute' => 'motivo',
    'format' => 'html',
    'value' => function ($model) {
        return
        \yii\helpers\Html::decode($model->motivoFechaPermiso->motivo);
    },
    'filter' => false,
    'options' => ['style' => 'width: 65%;'],
],
[
    'label' => 'Fecha de Permiso',
    'value' => function ($formato) {

        setlocale(LC_TIME, "es_419.UTF-8");
    }
];
```

```
$fechaPermiso =
strtotime($formato->motivoFechaPermiso->fecha_permiso);

$fechaFormatada = strftime('%A, %B %d, %Y',
$fechaPermiso);

setlocale(LC_TIME, null);

return $fechaFormatada;
},
],
];
break;

case 'PERMISO ECONOMICO':
$formatoAttributes = [
[
'label' => 'Motivo',
'attribute' => 'motivo',
'format' => 'html',
'value' => function ($model) {
    return
\yii\helpers\Html::decode($model->motivoFechaPermiso->motivo);
},
'filter' => false,
'options' => ['style' => 'width: 65%;'],
],
[
'label' => 'Fecha de Permiso',
'value' => function ($formato) {

    setlocale(LC_TIME, "es_419.UTF-8");

    $fechaPermiso =
strtotime($formato->motivoFechaPermiso->fecha_permiso);

    $fechaFormatada = strftime('%A, %B %d, %Y',
$fechaPermiso);

    setlocale(LC_TIME, null);

    return $fechaFormatada;
}
]
```

```
        },
    ],
];
break;

case 'CAMBIO DE DÍA LABORAL':
$formatoAttributes = [
[
    'label' => 'Motivo',
    'attribute' => 'motivo',
    'format' => 'html',
    'value' => function ($model) {
        return
\yii\helpers\Html::decode($model->motivoFechaPermiso->motivo);
    },
    'filter' => false,
    'options' => ['style' => 'width: 65%;'],
],
[
    'label' => 'Fecha de Permiso',
    'value' => function ($formato) {

        setlocale(LC_TIME, "es_419.UTF-8");

        $fechaPermiso =
strtotime($formato->motivoFechaPermiso->fecha_permiso);

        $fechaFormateada = strftime('%A, %B %d, %Y',
$fechaPermiso);

        setlocale(LC_TIME, null);

        return $fechaFormateada;
},
],
[
    'label' => 'Fecha a Laborar',
    'value' => function ($formato) {

        setlocale(LC_TIME, "es_419.UTF-8");
    }
];
```

```
$fechaPermiso =
strtotime($formato->fecha_a_laborar);

$fechaFormateada = strftime('%A, %B %d, %Y',
$fechaPermiso);

setlocale(LC_TIME, null);

return $fechaFormateada;
},
],
];
break;
case 'CAMBIO DE HORARIO DE TRABAJO':
$formatoAttributes = [
[
'label' => 'Motivo',
'attribute' => 'motivo',
'format' => 'html',
'value' => function ($model) {
    return
\yii\helpers\Html::decode($model->motivoFechaPermiso->motivo);
},
'filter' => false,
'options' => ['style' => 'width: 65%;'],
],
[
'label' => 'Fecha de Permiso',
'value' => function ($formato) {

    setlocale(LC_TIME, "es_419.UTF-8");

$fechaPermiso =
strtotime($formato->motivoFechaPermiso->fecha_permiso);

$fechaFormateada = strftime('%A, %B %d, %Y',
$fechaPermiso);

setlocale(LC_TIME, null);

return $fechaFormateada;
},
```

```
],
[
    'attribute' => 'horario_inicio',
    'value' => function ($formato) {
        $hora = date("g:i A",
        strtotime($formato->horario_inicio));
        return $hora;
    },
],
[
    'attribute' => 'horario_fin',
    'value' => function ($formato) {
        $hora = date("g:i A",
        strtotime($formato->horario_fin));
        return $hora;
    },
],
[
    'label' => 'Fecha de Inicio',
    'value' => function ($formato) {

        setlocale(LC_TIME, "es_419.UTF-8");

        $fechainicio= strtotime($formato->fecha_inicio);

        $fechaFormatada = strftime('%A, %B %d, %Y',
        $fechainicio);

        setlocale(LC_TIME, null);

        return $fechaFormatada;
    },
],
[
    'label' => 'Fecha de Termino',
    'value' => function ($formato) {

        setlocale(LC_TIME, "es_419.UTF-8");

        $fechatermino=
        strtotime($formato->fecha_termino);
    }
]
```

```
$fechaFormateada = strftime('%A, %B %d, %Y',
$fechaTermino);

setlocale(LC_TIME, null);

return $fechaFormateada;
},
],
];

break;

case 'PERMISO SIN GOCE DE SUELDO':
$formatoAttributes = [
[
'label' => 'Motivo',
'attribute' => 'motivo',
'format' => 'html',
'value' => function ($model) {
    return
\yii\helpers\Html::decode($model->motivoFechaPermiso->motivo);
},
'filter' => false,
'options' => ['style' => 'width: 65%;'],
],
[
'label' => 'Fecha de Permiso',
'value' => function ($formato) {

    setlocale(LC_TIME, "es_419.UTF-8");

    $fechaPermiso =
strtotime($formato->motivoFechaPermiso->fecha_permiso);

    $fechaFormateada = strftime('%A, %B %d,
%Y', $fechaPermiso);

    setlocale(LC_TIME, null);

    return $fechaFormateada;
},
],
];
```

```
];
break;

case 'CITA MEDICA':
$formatoAttributes = [
[

    'label' => 'Motivo de cita medica',
    'attribute' => 'comentario',
    'format' => 'html',
    'value' => function ($formato) {
        return
\yii\helpers\Html::decode($formato->comentario);
    },
    'filter' => false,
    'options' => ['style' => 'width: 65%;'],
],
[
    'label' => 'Fecha de Cita medica',
    'value' => function ($formato) {

        setlocale(LC_TIME, "es_419.UTF-8");

        $fechaPermiso =
strtotime($formato->fecha_para_cita);

        $fechaFormateada = strftime("%A, %B %d,
%Y', $fechaPermiso);

        setlocale(LC_TIME, null);

        return $fechaFormateada;
},
],
[
    'attribute' => 'horario_inicio',
    'value' => function ($formato) {
        $hora = date("g:i A",
strtotime($formato->horario_inicio));
        return $hora;
}
];
```

```
        },
    ],
    [
        'attribute' => 'horario_finalizacion',
        'value' => function ($formato) {
            $hora = date("g:i A",
strtoime($formato->horario_finalizacion));
            return $hora;
        },
    ],
];
break;
```

```
case 'CAMBIO DE PERIODO VACACIONAL':
$formatoAttributes = [
[
    'label' => 'Motivo',
    'value' => function ($formato) {
        return $formato->motivo;
    },
],
[
    'label' => 'Año',
    'value' => function ($formato) {
        return $formato->año;
    },
],
[
    'label' => 'Primera vez',
    'value' => function ($formato) {
        return $formato->primera_vez;
    },
],
[
    'label' => 'Numero de periodo',
    'value' => function ($formato) {
        return $formato->numero_periodo;
    },
],
```

```
[  
    'label' => 'Fecha de Inicio',  
    'value' => function ($formato) {  
  
        setlocale(LC_TIME, "es_419.UTF-8");  
  
        $fechalinicio=  
        strtotime($formato->fecha_inicio_período);  
  
        $fechaFormateada = strftime('%A, %B %d,  
        %Y', $fechalinicio);  
  
        setlocale(LC_TIME, null);  
  
        return $fechaFormateada;  
    },  
],  
[  
    'label' => 'Fecha de Fin',  
    'value' => function ($formato) {  
  
        setlocale(LC_TIME, "es_419.UTF-8");  
  
        $fechalinicio=  
        strtotime($formato->fecha_fin_período);  
  
        $fechaFormateada = strftime('%A, %B %d,  
        %Y', $fechalinicio);  
  
        setlocale(LC_TIME, null);  
  
        return $fechaFormateada;  
    },  
],  
];  
break;  
}  
  
echo DetailView::widget([  
    'model' => $formato,  
    'attributes' => $formatoAttributes,  
]);
```

```
    } else {
        echo '<p>No se encontró información del formato asociado.</p>';
    }

?>
    </div>
    <!--.col-md-12-->
</div>
<!--.row-->

    </div>
    <!--.card-body-->
</div>
<!--.card-->
</div>
<!--.col-md-6-->
</div>
<!--.row justify-content-center-->
</div>
<!--.container-fluid-->
```

### Formatos de incidencias.

En este caso se va a mostrar la vista de un solo formato (permiso\_fuera\_trabajo). El resto de formatos (comision\_especial, cambio\_dia\_laboral, cambio\_horario\_trabajo, permiso\_sin\_sueldo, permiso\_economico, cambio\_periodo\_vacacional, contrato\_personal\_eventual) siguen una logica y estructura muy similar en cuanto a sus vistas.

- \_form.

Descripción. Código del formulario que permite que el empleado pueda ingresar la solicitud necesaria para generar el registro que el archivo / formato va a recibir.

```
<?php
//IMPORTACIONES
use yii\helpers\Html;
use yii\bootstrap4\ActiveForm;
use yii\bootstrap5\Alert;
use froala\froalaeditor\FroalaEditorWidget;
use yii\web\View;
/* @var $this yii\web\View */
```

```
/* @var $model app\models\PermisoFueraTrabajo */
/* @var $form yii\bootstrap4\ActiveForm */

$this->registerCssFile('@web/css/site.css', ['position' => View::POS_HEAD]);

?>
<div class="container-fluid">
    <div class="row justify-content-center">
        <div class="col-md-10">
            <div class="card">
                <?php $form = ActiveForm::begin(['options' => ['id' => 'employee-form']]); ?>
                <div class="card-header bg-info text-white">
                    <h2>PERMISO FUERA DEL TRABAJO</h2>
                    <p> Empleado: <?= $empleado->nombre . ' ' . $empleado->apellido ?></p>

                <?php //SE OBTIENE EL USUARIO ACTIVO
                $usuarioActual = Yii::$app->user->identity;
                $empleadoActual = $usuarioActual->empleado;

                if ($empleadoActual->id === $empleado->id) {
                    echo Html::a('<i class="fa fa-home"></i> Inicio',
                    ['site/portalemppleado'], [
                        'class' => 'btn btn-outline-warning mr-3 float-right fa-lg',
                        'encode' => false,
                    ]);
                } else {
                    if (Yii::$app->user->can('crear-formatos-incidencias-empleados')) {
                        echo Html::a('<i class="fa fa-chevron-left"></i> Volver',
                        ['empleado/index'], [
                            'class' => 'btn btn-outline-warning mr-3 float-right fa-lg',
                            'encode' => false,
                        ]);
                    } else {
                        echo Html::a('<i class="fa fa-home"></i> Inicio',
                        ['site/portalemppleado'], [
                            'class' => 'btn btn-outline-warning mr-3 float-right fa-lg',
                            'encode' => false,
                        ]);
                    }
                }
            
```

```
?>

</div>
<div id="loading-spinner-laboral" style="display: none;">
    <i class="fa fa-spinner fa-spin fa-2x" style="color: #000000;"></i>
Procesando...
</div>
<div class="card-body">
    <div class="row">

        <div class="col-12">
            <p><strong>Permisos usados:</strong> <?= $permisosUsados
//MUESTRA LA CANTIDAD DE PERMISOS QUE HA SOLICITDAD
?></p>
            <p><strong>Permisos disponibles:</strong> <?=
$permisosDisponibles //MUESTRA LOS PERMISOS DISPONIBLES
?></p>
        </div>

        <div class="col-md-12">
            <div class="d-flex align-items-center mb-3">
                <?php
                //ALERTA
                foreach (Yii::$app->session->getAllFlashes() as $type =>
$message) {
                    if ($type === 'error') {
                        echo Alert::widget([
                            'options' => ['class' => 'alert-danger'],
                            'body' => $message,
                        ]);
                    } else {
                        echo Alert::widget([
                            'options' => ['class' => 'alert-' . $type],
                            'body' => $message,
                        ]);
                    }
                }
                ?>
            </div>

            <div class="permiso-fuera-trabajo-form">
```

```
<div class="card bg-light">

    <div class="card-body">
        <div class="row">

            <div class="col-6 col-sm-3">

                <?= $form->field($motivoFechaPermisoModel,
'fecha_permiso')->input('date')->label('Fecha de permiso') ?>
                </div>
                <div class="col-6 col-sm-2">

                    <?= $form->field($model,
' hora_salida')->input('time')->label('Hora de salida') ?>
                    </div>
                    <div class="col-6 col-sm-2">

                        <?= $form->field($model,
' hora_regreso')->input('time')->label('Hora de regreso') ?>
                        </div>
                        <div class="w-100"></div>
                        <div class="col-6 col-sm-3">

                            <?= $form->field($model,
' fecha_a_reponer')->input('date')->label('Fecha a reponer') ?>
                            </div>
                            <div class="col-6 col-sm-3">

                                <?= $form->field($model,
' horario_fecha_a_reponer')->input('time')->label('Horario de fecha a reponer') ?>
                                </div>
                                <?= $form->field($motivoFechaPermisoModel,
'motivo')->widget(FroalaEditorWidget::className(), [/WIDGET PARA
TRANSFORMAR EL INPUT EN UN INPUT CON EDITOR DE TEXTO
'options' => [
'id' => 'exp-fisca'
],
'clientOptions' => [
```

```
'toolbarInline' => false,  
'theme' => 'royal',  
'language' => 'es',  
'height' => 200,  
'pluginsEnabled' => [  
    'align',  
    'charCounter',  
    'codeBeautifier',  
    'codeView',  
    'colors',  
    'draggable',  
    'emoticons',  
    'entities',  
    'fontFamily',  
    'fontSize',  
    'fullscreen',  
    'inlineStyle',  
    'lineBreaker',  
    'link',  
    'lists',  
    'paragraphFormat',  
    'paragraphStyle',  
    'quickInsert',  
    'quote',  
    'save',  
    'table',  
    'url',  
    'wordPaste'  
]  
]  
])->label('Motivo:'); ?>  
  
<div class="col-6 col-sm-4">  
  
    </div>  
</div>  
  
<?= Html::submitButton('Generar <i class="fa  
fa-check"></i>', [  
    'class' => 'btn btn-success btn-lg float-right',
```

```
'id' => 'save-button-personal'
]) ?>
</div>
</div>

<?php ActiveForm::end(); ?>
<?php
$script = <<< JS
$('#employee-form').on('beforeSubmit', function() {

    var button = $('#save-button-personal');
    var spinner = $('#loading-spinner-laboral');

    button.prop('disabled', true); // Deshabilita el botón
    spinner.show(); // Muestra el spinner

    return true; // Permite que el formulario se envíe
});
JS;
$this->registerJs($script);
?>

</div>
</div>
</div>
</div>

</div>
</div>
</div>
</div>
```

- view.

Descripción. Código que muestra la información detallada del registro del formato que se genero.

```
<?php
//IMPORTACIONES
use yii\helpers\Html;
use yii\widgets\DetailView;
use yii\bootstrap5\Alert;
use app\models\Empleado;
use app\models\JuntaGobierno;
/* @var $this yii\web\View */
/* @var $model app\models\PermisoFueraTrabajo */

$this->title = $model->id;

\yii\web\YiiAsset::register($this);
?>

<div class="container-fluid">
<div class="row justify-content-center">
<div class="col-md-10">
    <div class="card">
        <div class="card-header bg-info text-white">
            <h2>PERMISO FUERA TRABAJO</h2>
            <p> Empleado: <?= $empleado->nombre.' .
'. $empleado->apellido ?></p>
        <?php
// Obtener el ID del usuario actual

$usuarioActual = Yii::$app->user->identity;
$empleadoActual = $usuarioActual->empleado;
$JuntaGobierno = JuntaGobierno::find()->where(['empleado_id' =>
$model->empleado_id])->one();

// Comparar el ID del empleado actual con el ID del empleado para el cual se
está creando o viendo el registro
if ($empleadoActual->id === $empleado->id) {
    // El empleado está viendo su propio registro
    echo Html::a('<i class="fa fa-home"></i> Inicio', ['site/portalemppleado'], [
        'class' => 'btn btn-outline-warning mr-3 float-right fa-lg',
        'encode' => false,
    ]);
}

// Verificar el nivel jerárquico del empleado para sus propios registros
if ($JuntaGobierno && ($JuntaGobierno->nivel_jerarquico === 'Jefe de
unidad' || $JuntaGobierno->nivel_jerarquico === 'Director')) {
    echo Html::a('<i class="fa fa-print" aria-hidden="true"></i> Vista Previa
de Impresión', ['export-html-segundo-caso', 'id' => $model->id], ['class' => 'btn
btn-success', 'target' => '_blank']);
}
```

```
    } else {
        echo Html::a('<i class="fa fa-print" aria-hidden="true"></i> Vista Previa
de Impresión', ['export-html', 'id' => $model->id], ['class' => 'btn btn-dark',
'target' => '_blank']);
    }
} else {
    // El empleado está creando o viendo un registro para otro empleado
    if (Yii::$app->user->can('crear-formatos-incidencias-empleados')) {
        echo Html::a('<i class="fa fa-chevron-left"></i> Volver',
['empleado/index'], [
        'class' => 'btn btn-outline-warning mr-3 float-right fa-lg',
        'encode' => false,
    ]);
    } else if (Yii::$app->user->can('gestor-rh')) {
        // Redirige al historial del navegador
        echo Html::a('<i class="fa fa-chevron-left"></i> Volver',
['javascript:void(0)'], [
        'class' => 'btn btn-outline-warning mr-3 float-right fa-lg',
        'encode' => false,
        'onclick' => 'window.history.back(); return false;',
    ]);
    }
}

else {
    echo Html::a('<i class="fa fa-home"></i> Inicio', ['site/portalemppleado'], [
        'class' => 'btn btn-outline-warning mr-3 float-right fa-lg',
        'encode' => false,
    ]);
}

// Verificar el nivel jerárquico del empleado para el cual se está creando el
registro
if ($juntaGobierno && ($juntaGobierno->nivel_jerarquico === 'Jefe de
unidad' || $juntaGobierno->nivel_jerarquico === 'Director')) {
    echo Html::a('<i class="fa fa-print" aria-hidden="true"></i> Vista Previa
de Impresión', ['export-html-segundo-caso', 'id' => $model->id], ['class' => 'btn
btn-success', 'target' => '_blank']);
} else {
    echo Html::a('<i class="fa fa-print" aria-hidden="true"></i> Vista Previa
de Impresión', ['export-html', 'id' => $model->id], ['class' => 'btn btn-dark',
'target' => '_blank']);
}

// Obtener el ID del usuario actual
$usuarioid = Yii::$app->user->identity->id;
```

```
// Obtener el empleado actual asociado al usuario
$empleado = Empleado::find()->where(['usuario_id' => $usuariold])->one();

if ($empleado) {
    // Asignar el ID del empleado al modelo
    $model->empleado_id = $empleado->id;

    // Obtener la Junta de Gobierno del empleado para el cual se está creando
    // el registro

    // Verificar el nivel jerárquico del empleado para el cual se está creando el
    // registro
    if ($juntaGobierno && ($juntaGobierno->nivel_jerarquico === 'Jefe de
    unidad' || $juntaGobierno->nivel_jerarquico === 'Director')) {
        // echo Html::a('<i class="fa fa-print" aria-hidden="true"></i> Vista Previa
        de Impresión', ['export-html-segundo-caso', 'id' => $model->id], ['class' => 'btn
        btn-success ', 'target' => '_blank']);
    } else {
        //echo Html::a('<i class="fa fa-print" aria-hidden="true"></i> Vista Previa
        de Impresión', ['export-html', 'id' => $model->id], ['class' => 'btn btn-dark ',
        'target' => '_blank']);
    }
} else {
    Yii::$app->session->setFlash('error', 'No se pudo encontrar el empleado
    asociado al usuario actual.');
    return $this->redirect(['index']);
}

?>

</div>
<div class="card-body">
    <div class="row">
        <div class="col-md-12">

            <?php
            //ALERTA
            foreach (Yii::$app->session->getAllFlashes() as $type => $message) {
                echo Alert::widget([
                    'options' => ['class' => 'alert-' . $type],

```

```
'body' => $message,
]);
}
?>
<?= DetailView::widget([
'model' => $model,
'attributes' => [
'solicitud_id',
[
'label' => 'Motivo',
'attribute' => 'motivo',
'format' => 'html',
'value' => function ($model) {
    return \yii\helpers\Html::decode($model->motivoFechaPermiso->motivo);
},
'filter' => false,
'options' => ['style' => 'width: 65%;'],
],
[
'label' => 'Hora de Salida',
'attribute' => 'hora_salida',
'value' => function ($model) {
    $hora = date("g:i A", strtotime($model->hora_salida));
    return $hora;
},
],
[
'label' => 'Hora de Regreso',
'attribute' => 'hora_regreso',
'value' => function ($model) {
    $hora = date("g:i A", strtotime($model->hora_regreso));
    return $hora;
},
],
[
'label' => 'Fecha que Repondrá',
'attribute' => 'fecha_a_reponer',
'value' => function ($model) {
    setlocale(LC_TIME, "es_419.UTF-8");
    $fechaAreponer = strtotime($model->fecha_a_reponer);
    $fechaFormateada = strftime('%A, %B %d, %Y', $fechaAreponer);
    setlocale(LC_TIME, null);
    return $fechaFormateada;
},
]
```

```
],
[
    'label' => 'Horario de fecha que Repondrá',
    'attribute' => 'horario_fecha_a_reponer',
    'value' => function ($model) {
        $hora = date("g:i A", strtotime($model->horario_fecha_a_reponer));
        return $hora;
    },
],
[
    'label' => 'Fecha de Permiso',
    'value' => function ($model) {
        setlocale(LC_TIME, "es_419.UTF-8");
        $fechaPermiso =
            strtotime($model->motivoFechaPermiso->fecha_permiso);
        $fechaFormatada = strftime('%A, %B %d, %Y', $fechaPermiso);
        setlocale(LC_TIME, null);
        return $fechaFormatada;
    },
],
])
?>

</div>
<!--.col-md-12-->
</div>
<!--.row-->
</div>
<!--.card-body-->
</div>
<!--.card-->
</div>
</div>
</div>
```

- historial / index.

Descripción. Este código se encarga de mostrar los registros que el empleado ha realizado del formato.

```
<?php
//IMPORTACIONES
use yii\helpers\Html;
use yii\grid\GridView;
```

```
use yii\widgets\Pjax;
use yii\web\View;
/* @var $this yii\web\View */
/* @var $searchModel app\models\PermisoFueraTrabajoSearch */
/* @var $dataProvider yii\data\ActiveDataProvider */

//SE CARGAN LOS ESTILOS
$this->registerCssFile('@web/css/site.css', ['position' =>
View::POS_HEAD]);
$this->registerCssFile('@web/css/grid-view.css', ['position' =>
View::POS_HEAD]);

$this->title = 'Permiso Fuera del Trabajo';

?>
<div class="container-fluid">
<div class="row justify-content-center">
<div class="col-md-10">
    <div class="card">
        <div class="card-header bg-info text-white">
            <h3>Historial de Permisos fuera de trabajo</h3>
            <p> Empleado: <?= $empleado->nombre.' .
'.>$empleado->apellido ?></p>
            <?php if
(Yii::$app->user->can('ver-empleados-departamento') ||
Yii::$app->user->can('ver-empleados-direccion') ) {?>
                <?= Html::a('<i class="fa fa-chevron-left"></i> Volver',
['empleado/index'], [
'class' => 'btn btn-outline-warning mr-3 float-right fa-lg',
'encode' => false, // Para que el HTML dentro del enlace no se escape
]) ?>

<?php } else if (Yii::$app->user->can('gestor-rh')) {
    // Redirige al historial del navegador
    echo Html::a('<i class="fa fa-chevron-left"></i> Volver',
'javascript:void(0)', [
'class' => 'btn btn-outline-warning mr-3 float-right fa-lg',
'encode' => false,
'onclick' => 'window.history.back(); return false;',
]);
} ?>
```

```
</div>
<div class="card-body">
    <div class="row mb-2">
        <div class="col-md-12">

            </div>
        </div>
    <?php Pjax::begin();?>

    <?= GridView::widget([ //SE MUESTRA LA LISTA DE LOS
        REGISTROS REALIZADOS
        'dataProvider' => $dataProvider,
        'filterModel' => $searchModel,
        //options' => ['class' => 'my-custom-grid-view'], // Clase específica
        para el GridView
        'columns' => [
            ['class' => 'yii\grid\SerialColumn'],

            [
                'attribute' => 'fecha_creacion',
                'label' => 'Fecha de creación',
                'value' => function ($model) {
                    return $model->solicitud->fecha_creacion;
                },
                // 'options' => ['style' => 'width: 30%;'],
            ],
            [
                'label' => 'Motivo',
                'attribute' => 'motivo',
                'format' => 'html',
                'value' => function ($model) {
                    return
                        \yii\helpers\Html::decode($model->motivoFechaPermiso->motivo);
                },
                'filter' => false,
                'options' => ['style' => 'width: 65%;'],
            ],
        ],
    ])
```

```
[ 'class' => 'hail812\adminlte3\yii\grid\ActionColumn',  
  
 //SI SOLO TIENE EL PERMISO DE VER, SOLO PODRA  
 OBSERVAR LOS REGISTROS, EN CAMBIO, SI NO, PUEDE  
 ELIMINAR LOS REGISTROS  
 'template' => Yii::$app->user->can('ver-empleados-departamento')  
 ? '{view}' : '{view} {delete}',  
  
 'buttons' => [  
  
     'view' => function ($url, $model) {  
         return Html::a('<i class="far fa-eye"></i>', $url, [  
             'title' => 'Ver solicitud',  
             'class' => 'btn btn-outline-info btn-sm',  
             'data-pjax' => "0"  
         ]);  
     },  
  
     'delete' => function ($url, $model) {  
         return Html::a('<i class="fas fa-trash"></i>', $url, [  
             'title' => Yii::t('yii', 'Eliminar'),  
             'data-confirm' => Yii::t('yii', '¿Estás seguro de que  
deseas eliminar este elemento?'),  
             'data-method' => 'post',  
             'class' => 'btn btn-outline-danger btn-sm'  
         ]);  
     },  
  
 ],  
 ],  
 'summaryOptions' => ['class' => 'summary mb-2'],  
 'pager' => [  
     'class' => 'yii\bootstrap4\LinkPager',  
 ],  
]) ?>
```

```
<?php Pjax::end(); ?>

</div>
<!--.card-body-->
</div>
<!--.card-->
</div>
<!--.col-md-12-->
</div>
<!--.row-->
</div>
```

- excel-html.

Descripción. Este código se encarga de generar una vista previa a la impresión, aquí se muestra el html generado del formato, cuenta con una serie de estilos que permite que se ajuste adecuadamente a la ventana de impresión del navegador.

```
<?php

use yii\helpers\Html;

$this->title = 'Permiso_Fuera_Trabajo_'. $model->empleado->nombre.'
.'.$model->empleado->apellido;
// $this->params['breadcrumbs'][] = $this->title;
?>
<head>
    <?php //IMPORTACIONES NECESARIAS?

        <!-- bootstrap 5.x or 4.x is supported. You can also use the bootstrap
        css 3.3.x versions -->
        <link rel="stylesheet"
        href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" crossorigin="anonymous">

        <!-- default icons used in the plugin are from Bootstrap 5.x icon library
        (which can be enabled by loading CSS below) -->
```

```
<link rel="stylesheet"
  href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.5.0/font/bootstrap-
  icons.min.css" crossorigin="anonymous">

<!-- alternatively you can use the font awesome icon library if using
with `fas` theme (or Bootstrap 4.x) by uncommenting below. -->
<!-- link rel="stylesheet"
  href="https://use.fontawesome.com/releases/v5.15.3/css/all.css"
  crossorigin="anonymous" -->

<!-- the fileinput plugin styling CSS file -->
<link
  href="https://cdn.jsdelivr.net/gh/kartik-v/bootstrap-fileinput@5.5.0/css/fil
  einput.min.css" media="all" rel="stylesheet" type="text/css" />

<!-- if using RTL (Right-To-Left) orientation, load the RTL CSS file after
fileinput.css by uncommenting below -->
<!-- link
  href="https://cdn.jsdelivr.net/gh/kartik-v/bootstrap-fileinput@5.5.0/css/fil
  einput-rtl.min.css" media="all" rel="stylesheet" type="text/css" /-->

<!-- the jQuery Library -->
<script src="https://code.jquery.com/jquery-3.6.0.min.js"
  crossorigin="anonymous"></script>

<!-- buffer.min.js and filetype.min.js are necessary in the order listed
for advanced mime type parsing and more correct
  preview. This is a feature available since v5.5.0 and is needed if
you want to ensure file mime type is parsed
  correctly even if the local file's extension is named incorrectly. This
will ensure more correct preview of the
  selected file (note: this will involve a small processing overhead in
scanning of file contents locally). If you
  do not load these scripts then the mime type parsing will largely be
derived using the extension in the filename
  and some basic file content parsing signatures. -->
<script
  src="https://cdn.jsdelivr.net/gh/kartik-v/bootstrap-fileinput@5.5.0/js/plug
  ins/buffer.min.js" type="text/javascript"></script>
<script
  src="https://cdn.jsdelivr.net/gh/kartik-v/bootstrap-fileinput@5.5.0/js/plug
  ins/filetype.min.js" type="text/javascript"></script>
```

```
<!-- piexif.min.js is needed for auto orienting image files OR when
restoring exif data in resized images and when you
wish to resize images before upload. This must be loaded before
fileinput.min.js -->
<script
src="https://cdn.jsdelivr.net/gh/kartik-v/bootstrap-fileinput@5.5.0/js/plugins/piexif.min.js" type="text/javascript"></script>

<!-- sortable.min.js is only needed if you wish to sort / rearrange files in
initial preview.
This must be loaded before fileinput.min.js -->
<script
src="https://cdn.jsdelivr.net/gh/kartik-v/bootstrap-fileinput@5.5.0/js/plugins/sortable.min.js" type="text/javascript"></script>

<!-- bootstrap.bundle.min.js below is needed if you wish to zoom and
preview file content in a detail modal
dialog. bootstrap 5.x or 4.x is supported. You can also use the
bootstrap js 3.3.x versions. -->
<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.1/dist/js/bootstrap.bundle.min.js" crossorigin="anonymous"></script>

<!-- the main fileinput plugin script JS file -->
<script
src="https://cdn.jsdelivr.net/gh/kartik-v/bootstrap-fileinput@5.5.0/js/fileinput.min.js"></script>

<!-- following theme script is needed to use the Font Awesome 5.x
theme ('fa5'). Uncomment if needed. -->
<!-- script
src="https://cdn.jsdelivr.net/gh/kartik-v/bootstrap-fileinput@5.5.0/themes/fa5/theme.min.js"></script -->

<!-- optionally if you need translation for your language then include
the locale file as mentioned below (replace LANG.js with your language
locale) -->
<script
src="https://cdn.jsdelivr.net/gh/kartik-v/bootstrap-fileinput@5.5.0/js/locales/LANG.js"></script>
<meta charset="<?= Yii::$app->charset ?>">
```

```
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width,
initial-scale=1">
<?php $this->registerCsrfMetaTags() ?>
<title><?= Html::encode($this->title) ?></title>
<?php $this->head() ?>
</head>
<style>
    @media print {
        .container-fluid, body * {
            visibility: hidden; /* Ocultar todos los elementos */
        }
        .excel-content, .excel-content * {
            visibility: visible; /* Hacer visible solo el área de impresión */
        }
        .print-area {
            position: absolute;
            top: 15%;

            left: 8%;
            width: 200%;
            max-height: 100%;
            transform: scale(1.10); /* Escalar el contenido al 75% */
            transform-origin: top left; /* Ajustar el origen de la
transformación */
        }
        /* Márgenes de la página */
        @page {
            size: auto;
            margin: 0;
        }
        /* Ajustar el contenido al área imprimible */
        body {
            margin: 0;
            padding: 0;
        }
    }
</style>
<div class="row justify-content-center">
```

```
<div class="excel-html-preview">
  <div class="print-area">
    <div class="excel-content">
      <?= $htmlContent //OBTIENE EL CONTENIDO DEL HTML
      QUE SE GENERA EN EL CONTROLADOR ?>
    </div>
  </div>
</div>

<style>
  .floating-btn {
    position: fixed;
    bottom: 45%; /* Ajusta la distancia desde la parte inferior */
    right: 30%; /* Ajusta la distancia desde la parte derecha */
    z-index: 1000; /* Asegúrate de que el botón esté por encima de
    otros elementos */
    padding: 10px 20px; /* Ajusta el relleno del botón */
    border-radius: 50px; /* Haz que el botón sea redondeado */
    box-shadow: 0 4px 8px rgba(0, 0, 0, 0.2); /* Añade una sombra
    para darle un efecto flotante */
  }
</style>

<!--.card-->

</div>
<?= Html::a('Imprimir', '#', ['class' => 'btn btn-primary no-print
floating-btn', 'onclick' => 'window.print(); return false;'])
//PERMITE ABRIR LA VENTANA DE IMPRESIÓN DEL NAVEGADOR
PARA IMPRIMIR EL CONTENIDO HTML, ES DECIR, EL FORMATO
GENERADO A FORMA DE HTML
?>
```

### Modelos.

A continuación se mostraran algunos de los modelos que contienen funciones o reglas importantes que se deben de tomar en cuenta, el resto de modelos tiene el mismo tipo de estructura habitual de un modelo.

- Empleado.

Descripción. En las reglas del modelo, se define qué atributos son obligatorios, algunas expresiones que deben de seguir algunos campos para que sean válidos, y los escenarios en que se utilizan estas reglas.

```
class Empleado extends \yii\db\ActiveRecord
{
    //ESCENARIO PARA IDENTIFICAR EN QUE MOMENTO SERAN
    OBLIGATORIOS ALGUNOS CAMPOS
    const SCENARIO_CREATE = 'create';
    const SCENARIO_UPDATE = 'update';
    const SCENARIO_UPDATE_INFO_EDU = 'update';

    /**
     * {@inheritDoc}
     */
    public static function tableName()
    {
        return 'empleado';
    }

    /**
     * {@inheritDoc}
     */
    public function rules()
    {
        return [
            ['numero_empleado', 'usuario_id', 'informacion_laboral_id', 'nombre',
            'apellido', 'email', 'profesion',
            ],
            'required'],
    }
}
```

```
[[{"numero_empleado": "usuario_id", "informacion_laboral_id": "cat_nivel_estudio_id", "parametro_formato_id": "edad", "codigo_postal": "expediente_medico_id"}, {"integer": [{"fecha_nacimiento": "safe"}, {"numero_empleado": "unique", "message": "Este número de empleado ya existe."}, {"numero_casa": "string", "max": 4}, {"estado_civil": "string", "max": 12}, {"nombre": "string", "max": 30}, {"nombre": "apellido", "match": "pattern": "/^a-zA-ZáéíóúÁÉÍÓÚññ\s+$/"}, {"message": "Solo se permiten letras y espacios."}], //EXPRESIÓN PARA VALIDAR NOMBRE  
[{"email": "email"}, {"apellido": "string", "max": 60}, {"email": "string", "max": 100}, {"foto": "string", "max": 255}, {"telefono": "telefono_contacto_emergencia"}, {"match": "pattern": "/^0-9+\-|(\()|\s)*$/"}, {"message": "El teléfono solo puede contener números, espacios y los símbolos +, -, (, )."}], //EXPRESIÓN PARA VALIDAR NUMERO DE CELULAR  
[{"telefono": "telefono_contacto_emergencia"}, {"string": "max": 15}, {"colonia": "string", "max": 50}, {"profesion": "string", "max": 15}, {"calle": "string", "max": 85}, {"nombre_contacto_emergencia": "string", "max": 90}, {"relacion_contacto_emergencia": "string", "max": 25}, {"curp": "match", "pattern": "/^([A-Z][AEIOUX][A-Z]{2})\d{2}(?:0[1-9]|1[0-2])(?:0[1-9]|1[0-2])(?:0[1-9]|1[0-2])\d{3}[01])[HM](?:AS|B[CS]|C[CLMSH]|D[FG]|G[TR]|HG|JC|M[CNS]|N[ETL]|OC|PL|Q[TR]|S[PLR]|T[CSL]|VZ|YN|ZS)([B-DF-HJ-NP-TV-Z]{3}[A-Z]\d)(\d)$/"}, {"message": "El CURP no tiene un formato válido."}], //EXPRESIÓN PARA VALIDAR CURP  
[{"curp": "string", "max": 18}, {"rfc": "match", "pattern": "/^([A-ZÑ&]{3,4}) ?(?:-)?(\d{2})(?:0[1-9]|1[0-2])(?:0[1-9]|1[0-2])(?:0[1-9]|1[0-2])\d{3}[01]) ?(?:- ?)?([A-Z]\d){2})([A]\d)\$/"}, {"message": "El RFC no tiene un formato válido."}], //EXPRESIÓN PARA VALIDAR RFC  
[{"rfc": "string", "max": 13}, {"nss": "match", "pattern": "/^\d{11}\$/"}, {"message": "El NSS debe contener exactamente 11 dígitos."}, {"nss": "string", "max": 11}]]
```

```
[['municipio', 'estado'], 'string', 'max' => 45],
[['cat_nivel_estudio_id'], 'exist', 'skipOnError' => true, 'targetClass' =>
CatNivelEstudio::class, 'targetAttribute' => ['cat_nivel_estudio_id' => 'id']]],

[['numero_empleado'], 'string', 'max' => 4],


[['institucion_educativa', 'titulo_grado'], 'string', 'max' => 65],
[['foto'], 'file', 'skipOnEmpty' => TRUE, 'extensions' => 'png, jpg, jpeg'],
[['expediente_medico_id'], 'exist', 'skipOnError' => true, 'targetClass' =>
ExpedienteMedico::class, 'targetAttribute' => ['expediente_medico_id' => 'id']]],

[['informacion_laboral_id'], 'exist', 'skipOnError' => true, 'targetClass' =>
InformacionLaboral::class, 'targetAttribute' => ['informacion_laboral_id' => 'id']]],

[['usuario_id'], 'exist', 'skipOnError' => true, 'targetClass' =>
Usuario::class, 'targetAttribute' => ['usuario_id' => 'id']],


[['numero_empleado', 'usuario_id', 'nombre', 'apellido', 'email',
'profesion'], 'required', 'on' => self::SCENARIO_CREATE],
[['fecha_nacimiento', 'sexo', 'estado_civil', 'curp', 'nss', 'rfc', 'colonia',
'calle', 'codigo_postal', 'numero_casa', 'telefono'], 'required', 'on' =>
self::SCENARIO_UPDATE],


];
}

public function scenarios()
{
    //SE ESTABLECEN LOS ESCENARIOS CON LOS CAMPOS QUE
    PARTICIPARAN EN CADA UNO DE ESTOS
    $scenarios = parent::scenarios();
    $scenarios[self::SCENARIO_CREATE] = ['numero_empleado',
    'usuario_id', 'nombre', 'apellido', 'email', 'profesion'];
    $scenarios[self::SCENARIO_UPDATE] = ['numero_empleado',
    'usuario_id', 'informacion_laboral_id', 'nombre', 'apellido', 'email',
    'profesion','fecha_nacimiento', 'sexo', 'estado_civil', 'curp', 'nss', 'rfc', 'colonia',
```

```
'calle', 'codigo_postal', 'numero_casa', 'telefono',
'telefono_contacto_emergencia'];
    return $scenarios;
}

/***
 * {@inheritDoc}
 */
public function attributeLabels()
{
    return [
        'id' => 'ID',
        'numero_empleado' => 'Numero de empleado',
        'usuario_id' => 'Usuario ID',
        'informacion_laboral_id' => 'Informacion Laboral ID',
        'cat_nivel_estudio_id' => 'Cat Nivel Estudio ID',
        'parametro_formato_id' => 'Parametro Formato ID',
        'nombre' => 'Nombre',
        'apellido' => 'Apellido',
        'fecha_nacimiento' => 'Fecha de nacimiento',
        'edad' => 'Edad',
        'sexo' => 'Sexo',
        'foto' => 'Foto',
        'telefono' => 'Telefono',
        'email' => 'Email',
        'estado_civil' => 'Estado Civil',
        'colonia' => 'Colonia',
        'calle' => 'Calle',
        'numero_casa' => 'Numero Casa',
        'codigo_postal' => 'Codigo Postal',
        'nombre_contacto_emergencia' => 'Nombre',
        'relacion_contacto_emergencia' => 'Parentesco',
        'telefono_contacto_emergencia' => 'Telefono',
        'institucion_educativa' => 'Institucion Educativa',
        'titulo_grado' => 'Titulo Grado',
        'profesion' => 'Profesion',
        'curp' => 'CURP',
        'rfc' => 'RFC',
        'nss' => 'NSS',
        'municipio' => 'Municipio',
        'estado' => 'Estado',
```

```
'expediente_medico_id' => 'Expediente Medico ID',  
  
];  
}  
  
/**  
 * Gets query for [[Documentos]].  
 *  
 * @return \yii\db\ActiveQuery  
 */  
  
// RELACIONES  
public function getDocumentos()  
{  
    return $this->hasMany(Documento::class, ['empleado_id' => 'id']);  
}  
  
/**  
 * Gets query for [[InformacionLaboral]].  
 *  
 * @return \yii\db\ActiveQuery  
 */  
public function getInformacionLaboral()  
{  
    return $this->hasOne(InformacionLaboral::class, ['id' =>  
        'informacion_laboral_id']);  
}  
  
/**  
 * Gets query for [[Usuario]].  
 *  
 * @return \yii\db\ActiveQuery  
 */  
public function getUsuario()  
{  
    return $this->hasOne(Usuario::class, ['id' => 'usuario_id']);  
}
```

```
public function getCatNivelEstudio()
{
    return $this->hasOne(CatNivelEstudio::class, ['id' =>
'cat_nivel_estudio_id']);
}
public function getExpedienteMedico()
{
    return $this->hasOne(ExpedienteMedico::class, ['id' =>
'expediente_medico_id']);
}

}
```

- Documento.

Descripción. En las reglas del modelo, se define qué atributos son obligatorios, algunas expresiones que deben de seguir algunos campos para que sean válidos, y la función que permite la subida de archivos y se guarde en los directorios definidos.

```
class Documento extends \yii\db\ActiveRecord
{
    const SCENARIO_NOMBRE = 'nombre';

    /**
     * {@inheritDoc}
     */

    public $archivoSubido;

    public static function tableName()
    {
        return 'documento';
    }

    /**
     * {@inheritDoc}
     */
    public function rules()
    {
        return [

```

```
        [['empleado_id', 'ruta'], 'required'],
        [['empleado_id', 'cat_tipo_documento_id'], 'integer'],
        [['fecha_subida'], 'safe'],
        [['nombre'], 'string', 'max' => 85],
        [['ruta'], 'string', 'max' => 255],
        [['archivoSubido'], 'file', 'skipOnEmpty' => true, 'extensions' => 'png, jpg, pdf,
doc, docx'],//EXTENSIONES PERMITIDAS DE DOCUMENTOS
        [['observacion'], 'string'],

        [['empleado_id'], 'exist', 'skipOnError' => true, 'targetClass' =>
Empleado::class, 'targetAttribute' => ['empleado_id' => 'id']],
        [['cat_tipo_documento_id'], 'exist', 'skipOnError' => true, 'targetClass' =>
CatTipoDocumento::class, 'targetAttribute' => ['cat_tipo_documento_id' => 'id']],
    ];
}

/** 
 * {@inheritDoc}
 */
public function attributeLabels()
{
    return [
        'id' => 'ID',
        'empleado_id' => 'Empleado ID',
        'cat_tipo_documento_id' => 'Cat Tipo Documento ID',
        'nombre' => 'Nombre',
        'ruta' => 'Archivo',
        'fecha_subida' => 'Fecha Subida',
        'archivoSubido' => 'Archivo a Subir',
        'observacion' => 'Observaciones'

    ];
}
//RELACIONES
/**
 * Gets query for [[CatTipoDocumento]].
 *
 * @return \yii\db\ActiveQuery
 */
public function getCatTipoDocumento()
{
```

```
return $this->hasOne(CatTipoDocumento::class, ['id' =>
'cat_tipo_documento_id']);
}

/**
 * Gets query for [[Empleado]].
 *
 * @return \yii\db\ActiveQuery
 */
public function getEmpleado()
{
    return $this->hasOne(Empleado::class, ['id' => 'empleado_id']);
}

//FUNCION QUE PERMITE LA SUBIDA DE ARCHIVOS EN EL DIRECTORIO DE
RUNTIME, A SU VES SIGUE LA
//LOGICA NECESARIA PARA GUARDAR EL ARCHIVO EN LA CARPETA DEL
EMPLEADO
public function upload()
{
    if ($this->validate()) {
        $empleado = Empleado::findOne($this->empleado_id);
        $nombreCarpetaTrabajador = Yii::getAlias('@runtime') . '/empleados/' .
$empleado->nombre . '_' . $empleado->apellido;
        $nombreCarpetaExpedientes = $nombreCarpetaTrabajador . '/documentos';
        $rutaArchivo = $nombreCarpetaExpedientes . '/';
$this->archivoSubido->baseName . '' . $this->archivoSubido->extension;
        $this->archivoSubido->saveAs($rutaArchivo);
        $this->ruta = $rutaArchivo;
        return true;
    } else {
        return false;
    }
}

}
```

- CambioHorarioTrabajo.

Descripción. En las reglas del modelo, se define qué atributos son obligatorios, algunas expresiones que deben de seguir algunos campos para que sean válidos. También se establece una función que permite recibir

fechas en rango y otra función que valida la cantidad permisos anuales (establecidos en site/configuración) que tiene permitidos realizar en base al formato. La lógica que maneja la función que valida la cantidad de registros anuales es similar en cada uno de los modelos de los formatos.

```
class CambioHorarioTrabajo extends \yii\db\ActiveRecord
{
    public $jefe_departamento_id;
    public $dateRange;

    /**
     * {@inheritDoc}
     */
    public static function tableName()
    {
        return 'cambio_horario_trabajo';
    }

    /**
     * {@inheritDoc}
     */
    public function rules()
    {
        return [
            [['empleado_id', 'solicitud_id', 'motivo_fecha_permiso_id'], 'integer'],
            [['horario_inicio', 'horario_fin', 'fecha_inicio', 'fecha_termino'], 'safe'],
            [['horario_inicio', 'turno', 'horario_fin'], 'required'],
            [['turno'], 'string', 'max' => 12],
            [['nombre_jefe_departamento'], 'string', 'max' => 90],
            [['empleado_id'], 'exist', 'skipOnError' => true, 'targetClass' =>
                Empleado::class, 'targetAttribute' => ['empleado_id' => 'id']],
            [['motivo_fecha_permiso_id'], 'exist', 'skipOnError' => true, 'targetClass' =>
                MotivoFechaPermiso::class, 'targetAttribute' => ['motivo_fecha_permiso_id' => 'id']],
            [['solicitud_id'], 'exist', 'skipOnError' => true, 'targetClass' => Solicitud::class,
            'targetAttribute' => ['solicitud_id' => 'id']],
            [['jefe_departamento_id'], 'safe'],
            [['dateRange'], 'required', 'message' => 'El rango de fechas no puede estar
            vacío.'], // Regla y mensaje personalizado para dateRange
            ['empleado_id', 'validarLimiteAnual'], // VALIDA EL LIMITE ANUAL
            ESTABLECIDO
```

```
];
}

/**
 * {@inheritDoc}
 */
public function attributeLabels()
{
    return [
        'id' => 'ID',
        'empleador_id' => 'Empleador ID',
        'solicitud_id' => 'Solicitud ID',
        'motivo_fecha_permiso_id' => 'Motivo Fecha Permiso ID',
        'turno' => 'Turno',
        'horario_inicio' => 'Inicio de horario',
        'horario_fin' => 'Fin de horario',
        'fecha_inicio' => 'Fecha Inicio',
        'fecha_termino' => 'Fecha Termino',
        'nombre_jefe_departamento' => 'Nombre Jefe Departamento',
        'jefe_departamento_id' => 'Jefe de Departamento',
        'dateRange' => 'Rango de Fechas',
    ];
}

/**
 * Gets query for [[Empleador]].
 *
 * @return \yii\db\ActiveQuery
 */
public function getEmpleador()
{
    return $this->hasOne(Empleador::class, ['id' => 'empleador_id']);
}

/**
 * Gets query for [[MotivoFechaPermiso]].
 *
 * @return \yii\db\ActiveQuery
 */
public function getMotivoFechaPermiso()
{
```

```
    return $this->hasOne(MotivoFechaPermiso::class, ['id' =>
'motivo_fecha_permiso_id']);
}

/**
 * Gets query for [[Solicitud]].
 *
 * @return \yii\db\ActiveQuery
 */
public function getSolicitud()
{
    return $this->hasOne(Solicitud::class, ['id' => 'solicitud_id']);
}

// FUNCIÓN QUE PERMITE RECIBIR FECHAS EN RANGO DESDE INPUTS DE
DATE RANGE
public function beforeSave($insert)
{
    if (parent::beforeSave($insert)) {
        if ($this->dateRange) {
            list($this->fecha_inicio, $this->fecha_termino) = explode(' a ',
$this->dateRange);
        }

        return true;
    } else {
        return false;
    }
}

//FUNCION QUE VALIDA EL LIMITE DE REGISTROS ANUALES,
//SE IDENTIFICA EL AÑO PRESENTE, EL TIPO DE PERMISO Y EL TIPO DE
CONTRATO DEL EMPLEADO
//Y SABER LA CANTIDAD DE PERMISOS QUE TIENE SU TIPO DE CONTRATO
public function validarLimiteAnual($attribute, $params)
{
    $añoActual = date('Y');
    $empleado = Empleado::findOne($this->empleado_id);
    $tipoContratoid = $empleado->informacionLaboral->cat_tipo_contrato_id;

    // Contar los registros de ComisionEspecial del año actual, basándose en la
fecha de MotivoFechaPermiso
```

```
$contadorPermisos = static::find()
    ->joinWith('motivoFechaPermiso') // Unir con la tabla motivo_fecha_permiso
    ->where(['empleado_id' => $this->empleado_id])
    ->andWhere(['between', 'motivo_fecha_permiso.fecha_permiso',
"$añoActual-01-01", "$añoActual-12-31"])
    ->count();

$limiteAnual = ParametroFormato::find()
    ->where(['tipo_permiso' => 'CAMBIO DE HORARIO DE TRABAJO',
'cat_tipo_contrato_id' => $tipoContratold])
    ->one()->limite_anual;

if ($contadorPermisos >= $limiteAnual) {
    $this->addError($attribute, 'Has alcanzado el límite anual de cambio de
horario de trabajo para tu tipo de contrato.');
}
}
```

- Usuario.

Descripción. En las reglas del modelo, se define qué atributos son obligatorios, algunas expresiones que deben de seguir algunos campos para que sean válidos. Aquí se verifica que tipo de usuarios son los que se manejan en el sistema, validando roles, estatus, y credenciales del usuario al iniciar sesión.

```
<?php
//IMPORTACIONES
namespace app\models;

use Yii;

/**
 * This is the model class for table "usuario".
 *
 * @property int $id
 * @property string $password
 * @property int $status
 * @property int $rol
 * @property int $nuevo
 * @property string $username
*
```

```
* @property Empleado[] $empleados
*/
class Usuario extends \yii\db\ActiveRecord implements
\yii\web\IdentityInterface
{
    const SCENARIO_CREATE = 'create';
    /**
     * {@inheritDoc}
     */
    public static function tableName()
    {
        return 'usuario';
    }

    /**
     * {@inheritDoc}
     */
    public function rules()
    {
        return [
            [['username', 'status', 'rol'], 'required'],
            [['status', 'rol', 'nuevo'], 'integer'],
            [['username'], 'string', 'max' => 30],
            [['password'], 'string', 'max' => 64],
            [['username', 'password', 'status'], 'required', 'on'=> self::SCENARIO_CREATE],
        ];
    }

    /**
     * {@inheritDoc}
     */
    public function attributeLabels()
    {
        return [
            'id' => 'ID',
            'password' => 'Contraseña',
            'status' => 'Status',
            'rol' => 'Rol',
            'nuevo' => 'Nuevo',
            'username' => 'Usuario',
        ];
    }
}
```

```
];
}

//FUNCIONES PARA VALIDAR EL ROL DEL TIPO DE USUARIO Y SU
ESTATUS
public static function isUserAdmin($id)
{
    $user = self::findOne(['id' => $id, 'status' => '10']);
    return $user !== null && $user->rol === 2;
}

public static function isUserSimple($id)
{
    $user = self::findOne(['id' => $id, 'status' => '10']);
    return $user !== null && $user->rol === 1;
}

public static function isUserMedico($id)
{
    $user = self::findOne(['id' => $id, 'status' => '10']);
    return $user !== null && $user->rol === 3;
}

public static function isUserNuevo($id)
{
    $user = self::findOne(['id' => $id, 'status' => '10', 'nuevo' => '4']);
    return $user !== null && $user->rol === 1 && $user->nuevo !=4;
}

public function getAuthKey() {
    return null;
    //return $this->auth_key;
}
```

```
// interface
public function validateAuthKey($authKey) {
    return $this->getAuthKey() == $authKey;
}

// interface
public static function findIdentityByAccessToken($token, $type = null) {
    throw new
\yii\base\NotSupportedException("findIdentityByAccessToken' is not
implemented");
}

/* Identity Interface */
public function getId(){
    return $this->id;
}

// FUNCIONES QUE SE ENCARGAN DE BUSCAR Y VALIDAR QUE LAS
CREENCIAS DEL USUARIO INGRESADO
//EXISTEN EN LA BAE DE DATOS
public static function findIdentity($id) {
    return static::findOne(['id' => $id]);
}

public static function findByUserName($username) {
    return static::findOne(['username' => $username]);
}

public function validatePassword($password) {
    return Yii::$app->security->validatePassword($password,
$this->password);
}

/**
 * Gets query for [[Empleados]].
 *
 * @return \yii\db\ActiveQuery
 */

//RELACIONES CON LA TABLA EMPLEADO
public function getEmpleados()
{
```

```
        return $this->hasMany(Empleado::class, ['usuario_id' => 'id']);
    }

    public function getEmpleado()
{
    return $this->hasOne(Empleado::class, ['usuario_id' => 'id']);
}

}
```

- LoginForm.  
Descripción. En las reglas del modelo, se define qué atributos son obligatorios, algunas expresiones que deben de seguir algunos campos para que sean válidos. Aquí se validan las credenciales cuando los usuarios inician sesión, verifica que existe el registro del usuario, que la contraseña sea la correcta y si el estado del usuario es activo o inactivo.

```
<?php
// IMPORTACIONES
namespace app\models;

use Yii;
use yii\base\Model;

/**
 * LoginForm is the model behind the login form.
 *
 * @property-read User|null $user
 */
class LoginForm extends Model
{
    public $username;
    public $password;
    public $rememberMe = true;

    private $_user = false;
```

```
/**  
 * @return array the validation rules.  
 */  
public function rules()  
{  
    return [  
        ['username', 'password'], 'required',  
        ['rememberMe', 'boolean'],  
        ['password', 'validatePassword'],  
    ];  
}  
public function attributeLabels()  
{  
    return [  
  
        'password' => 'Contraseña',  
  
        'username' => 'Usuario',  
  
        'rememberMe' => 'Recuerdame'  
    ];  
}  
/**  
 * Validates the password.  
 * This method serves as the inline validation for password.  
 *  
 * @param string $attribute the attribute currently being validated  
 * @param array $params the additional name-value pairs given in the rule  
 */  
  
/// FUNCION PARA VALIDAR LA CONTRASEÑA  
public function validatePassword($attribute, $params)  
{  
    if (!$this->hasErrors()) {  
        $user = $this->getUser();  
  
        // if(!$user || $user->status === 0 ){  
        //     $this->addError($attribute, 'Usuario inactivo');  
    }  
}
```

```
//}
if(!$user){
    $this->addError($attribute, 'Usuario no encontrado.');
}
elseif( $user->status === 0){
    $this->addError($attribute, 'Usuario inactivo.');

}

else
{
    if(!$user->validatePassword($this->password))
        $this->addError($attribute, 'Usuario o Contraseña incorrecta.');
    //echo "$user->status";
    echo '<script>console.log("'.$user->status.'");</script>';

    // $this->addError($attribute, 'usuario activo.');
}

}

/**
 * Logs in a user using the provided username and password.
 * @return bool whether the user is logged in successfully
 */
//FUNCION QUE VALIDA EL USUARIO
public function login()
{
    $usuario = $this->getUser();
    if ($this->validate()) {
        if($usuario){
            return Yii::$app->user->login($this->getUser(), $this->rememberMe ?
3600*24*30 : 0);
        }
    }
    return false;
}

/**
 * Finds user by [[username]]
 *
```

```
* @return User|null
*/
// FUNCION QUE OBTIENE Y VERIFICA SI EXISTE REGISTRO DE LAS
// CREDENCIALES INGRESADAS POR EL USUARIO
public function getUser()
{
    if ($this->_user === false) {
        $this->_user = Usuario::findByUsername($this->username);
    }

    return $this->_user;
}
}
```

## Controladores.

A continuación se mostrarán algunas de las funciones (Actions) que brindan las principales funcionalidades del sistema. El resto de funciones en los controladores tienen la misma estructura y lógica básica que brinda el generador de código de yii2.

- Creación del empleado.

Descripción. En esta función se reciben los datos ingresados en el formulario (\_form) de empleado. Aquí se crea el registro y se maneja toda la lógica de las relaciones establecidas. Es decir, cuando se crea el empleado también se le asigna un registro de información laboral, vacaciones, periodos vacacionales, expediente médico. Además, al crear el empleado también se crea el usuario del empleado, con su respectivo rol y las credenciales se envían al email del empleado. En este código también se maneja la lógica que crea la estructura de carpetas del empleado que se creó, es decir, en el directorio runtime, se crea una carpeta especial para el empleado con una serie de subdirectorios donde se guardarán documentos del empleado, entre otros. Otra parte importante del código, es que llama unas funciones que asignan la cantidad de vacaciones que se asignan al empleado, esta información se guarda el registro de vacaciones y periodos vacacionales que son asociados al empleado. Estas funciones se encargan de calcular las vacaciones que se le asignan al empleado en base a su tipo de contrato y la fecha en que ingresó a la empresa.

```
//FUNCION PARA CREAR EL EMPLEADO Y SU USUARIO, TAMBIEN LOS
//REGISTROS ASOCIADOS
public function actionCreate()
{
```

```
//SE CARGAN LOS MODELOS ASOCIADOS AL EMPLEADO
$model = new Empleado();
$usuario = new Usuario();
$informacion_laboral = new InformacionLaboral();
$vacaciones = new Vacaciones();
$periodoVacacional = new PeriodoVacacional();
$segundoPeriodoVacacional = new SegundoPeriodoVacacional();
$juntaGobiernoModel = new JuntaGobierno();
$expedienteMedico = new ExpedienteMedico();
$antecedenteHereditario = new AntecedenteHereditario();

$usuario->scenario = Usuario::SCENARIO_CREATE;
$model->scenario = Empleado::SCENARIO_CREATE;

if ($model->load(Yii::$app->request->post()) &&
$usuario->load(Yii::$app->request->post()) &&
$informacion_laboral->load(Yii::$app->request->post())) {
    //INICIAMOS LA TRANSACCIÓN
    $transaction = Yii::$app->db->beginTransaction();
    $usuario->username = $model->nombre . $model->apellido;
    $nombres = explode(" ", $model->nombre);
    $apellidos = explode(" ", $model->apellido);
    $usernameBase = strtolower($nombres[0][0] . $apellidos[0] .
(isset($apellidos[1]) ? $apellidos[1][0] : ""));
    //SE GENERA EL NOMBRE DE USUARIO QUE TENDRA EL
    //EMPLEADO EN BASE A SU NOMBRE Y APELLIDO
    $usuario->username = $usernameBase;
    $counter = 1;

    while (Usuario::find()->where(['username' =>
$usuario->username])->exists()) {
        $usuario->username = $usernameBase . $counter;
        $counter++;
    }
    //SE ESTABLECE LA CONTRASEÑA POR DEFECTO
    $usuario->password = 'contrasena';
    $usuario->status = 10;
    $usuario->nuevo = 4;
    $hash =
Yii::$app->security->generatePasswordHash($usuario->password);
    $usuario->password = $hash;
```

```
try {
    // Calcula los días de vacaciones y guarda la información laboral
    $totalDiasVacaciones =
$this->calcularDiasVacaciones($informacion_laboral->fecha_ingreso,
$informacion_laboral->cat_tipo_contrato_id);
$diasPorPeriodo = ceil($totalDiasVacaciones / 2);

//GUARDA LA INFORMACION DE LOS PERIODOS VACAIONALES
$periodoVacacional->dias_vacaciones_periodo = $diasPorPeriodo;
if (!$periodoVacacional->save()) {
    Yii::$app->session->setFlash('error', 'Error al guardar
PeriodoVacacional.');
    throw new \yii\db\Exception('Error al guardar PeriodoVacacional');
}

$segundoPeriodoVacacional->dias_vacaciones_periodo =
$totalDiasVacaciones - $diasPorPeriodo;
if (!$segundoPeriodoVacacional->save()) {
    Yii::$app->session->setFlash('error', 'Error al guardar
SegundoPeriodoVacacional.');
    throw new \yii\db\Exception('Error al guardar
SegundoPeriodoVacacional');
}

$vacaciones->periodo_vacacional_id = $periodoVacacional->id;
$vacaciones->segundo_periodo_vacacional_id =
$segundoPeriodoVacacional->id;
$vacaciones->total_dias_vacaciones = $totalDiasVacaciones;
if (!$vacaciones->save()) {
    Yii::$app->session->setFlash('error', 'Error al guardar Vacaciones.');
    throw new \yii\db\Exception('Error al guardar Vacaciones');
}

//IDENTIFICA EL DEPARTAMENTO QUE SE MANDO Y ASIGNA A
INFORMACION LABORAL LA DIRECCION Y DPTO AL QUE PERTENECE
$departamento =
CatDepartamento::findOne($informacion_laboral->cat_departamento_id);
if ($departamento) {
    $informacion_laboral->cat_direccion_id =
$departamento->cat_direccion_id;
```

```
$informacion_laboral->cat_dpto_cargo_id =
$departamento->cat_dpto_id;
}

$informacion_laboral->vacaciones_id = $vacaciones->id;
if (!$informacion_laboral->save()) {
    Yii::$app->session->setFlash('error', 'Error al guardar
InformacionLaboral.');
    throw new \yii\db\Exception('Error al guardar InformacionLaboral');
}

//VERIFICA Y GUARDA EL ROL QUE SE ESTABLECIO
$model->informacion_laboral_id = $informacion_laboral->id;
if ($usuario->save()) {
    $model->usuario_id = $usuario->id;

    if ($usuario->rol == 1) {
        $rol = 'empleado';
    } elseif ($usuario->rol == 2) {
        $rol = 'gestor-rh';
    } elseif ($usuario->rol == 3) {
        $rol = 'medico';
    } else {
        $rol = 'empleado';
    }

    $auth = Yii::$app->authManager;
    $authorRole = $auth->getRole($rol);
    if ($authorRole) {
        $auth->assign($authorRole, $usuario->id);
    } else {
        Yii::$app->session->setFlash('error', "El rol $rol no existe en el
sistema de RBAC.");
        throw new \Exception("El rol $rol no existe en el sistema de
RBAC.");
    }

    // Subir foto del empleado si se proporcionó
    ///RECIBE LA FOTO DEL EMPLEADO, ADEMÁS SE CREAN LAS
    CARPETAS DEL EMPLEADO EN RUNTIME
    $upload = UploadedFile::getInstance($model, 'foto');
    if (is_object($upload)) {
```

```
$nombreCarpetaTrabajador = Yii::getAlias('@runtime') .  
'/empleados/' . $model->nombre . '_' . $model->apellido;  
if (!is_dir($nombreCarpetaTrabajador)) {  
    mkdir($nombreCarpetaTrabajador, 0775, true);  
}  
$nombreCarpetaUsuarioProfile = $nombreCarpetaTrabajador .  
'/foto_empleado';  
if (!is_dir($nombreCarpetaUsuarioProfile)) {  
    mkdir($nombreCarpetaUsuarioProfile, 0775, true);  
}  
  
$nombreIncidenciasCarpeta = $nombreCarpetaTrabajador .  
'/solicitudes_incidencias_empleado';  
if (!is_dir($nombreIncidenciasCarpeta)) {  
    mkdir($nombreIncidenciasCarpeta, 0775, true);  
}  
$upload_filename = $nombreCarpetaUsuarioProfile . '/' .  
$upload->baseName . '.' . $upload->extension;  
$upload->saveAs($upload_filename);  
$model->foto = $upload_filename;  
}  
  
// Asignar primera_revision a 0 antes de guardar el expediente  
médico  
$expedienteMedico->antecedente_hereditario_id =  
$antecedenteHereditario->id;  
$expedienteMedico->primera_revision = 0; // Establecer  
primera_revision a 0  
  
if ($expedienteMedico->save()) {  
    $model->expediente_medico_id = $expedienteMedico->id;  
  
    if ($model->save()) {  
        // Determinar el nivel jerárquico basado en el puesto  
        $puesto =  
CatPuesto::findOne($informacion_laboral->cat_puesto_id);  
        if ($puesto) {  
            $nivelJerarquico = 'Comun';  
            if (stripos($puesto->nombre_puesto, 'DIRECTOR') !== false)  
{  
                $nivelJerarquico = 'Director';  
            }  
        }  
    }  
}
```

```
        } elseif (stripos($puesto->nombre_puesto, 'JEFE DE
UNIDAD') !== false) {
            $nivelJerarquico = 'Jefe de unidad';
        } elseif (stripos($puesto->nombre_puesto, 'JEFE DEL
DEPARTAMENTO') !== false) {
            $nivelJerarquico = 'Jefe de departamento';
        }

        // Guardar la información en junta_gobierno si aplica
        if ($nivelJerarquico !== 'Comun') {
            $juntaGobiernoModel->empleado_id = $model->id;
            $juntaGobiernoModel->nivel_jerarquico =
            $nivelJerarquico;
            $juntaGobiernoModel->cat_departamento_id =
            $informacion_laboral->cat_departamento_id;
            $juntaGobiernoModel->cat_direccion_id =
            $informacion_laboral->cat_direccion_id; // Asigna la dirección correcta
            if (!$juntaGobiernoModel->save()) {
                Yii::$app->session->setFlash('error', 'Error al guardar
                JuntaGobierno.');
                $transaction->rollBack();
                throw new \yii\db\Exception('Error al guardar
                JuntaGobierno: ' . json_encode($juntaGobiernoModel->errors));
            }
        }

        // MANDA AL CORREO DEL EMPLEADO LAS
        CREDENCIALES PARA INICIAR SESION
        Yii::$app->mailer->compose()
            ->setFrom('elitaev7@gmail.com')
            ->setTo($model->email)
            ->setSubject('Datos de acceso al sistema')
            ->setTextBody("Nos comunicamos con usted,
{$model->nombre}\nAquí están tus datos de acceso:\nUsuario:
{$usuario->username}\nContraseña: contrasena")
            ->send();

        $transaction->commit();
        Yii::$app->session->setFlash('success', "Se guardó la
        información correctamente.");
        return $this->redirect(['view', 'id' => $model->id]);
    } else {
```

```
        Yii::$app->session->setFlash('error', 'Error al guardar  
Empleado.);  
        throw new \yii\db\Exception('Error al guardar Empleado');  
    }  
} else {  
    // Yii::$app->session->setFlash('error', 'Error al guardar  
ExpedienteMedico.);  
    // throw new \yii\db\Exception('Error al guardar  
ExpedienteMedico');  
}  
}  
} else {  
    //Yii::$app->session->setFlash('error', 'Error al guardar  
AntecedenteHereditario.);  
    throw new \yii\db\Exception('Error al guardar  
AntecedenteHereditario');  
}  
}  
} else {  
    Yii::$app->session->setFlash('error', 'Error al guardar Usuario.);  
    throw new \yii\db\Exception('Error al guardar Usuario');  
}  
}  
}  
  
    return $this->render('create', [  
//RENDERIZA LA VISTA, CARGANDO LOS MODELOS  
CORRESPONDIENTES  
    'model' => $model,  
    'usuario' => $usuario,  
    'informacion_laboral' => $informacion_laboral,  
    'vacaciones' => $vacaciones,  
    'periodoVacacional' => $periodoVacacional,  
    'segundoPeriodoVacacional' => $segundoPeriodoVacacional,  
    'expedienteMedico' => $expedienteMedico,  
    'antecedenteHereditario' => $antecedenteHereditario,  
    'puestos' => CatPuesto::find()->all(),  
    'departamentos' => CatDepartamento::find()->all(),  
    'juntaGobiernoModel' => $juntaGobiernoModel,  
]);
```

}

//FUNCION QUE DEFINE EL DIA DE VACACIONES EN BASE AL NOMBRE DEL TIPO DE FORMATO

```
private function calcularDiasVacaciones($fechalngreso, $tipoContratold)
{
    $fechalngreso = new \DateTime($fechalngreso);
    $fechaActual = new \DateTime();
    $intervalo = $fechalngreso->diff($fechaActual);
    $aniosTrabajados = $intervalo->y;

    $tipoContrato = CatTipoContrato::findOne($tipoContratold);

    if (!$tipoContrato) {
        return 0;
    }

    switch ($tipoContrato->nombre_tipo) {
        case 'Eventual':
            return $this->calcularVacacionesEventual($aniosTrabajados);
        case 'Confianza':
            return $this->calcularVacacionesConfianza($aniosTrabajados);
        case 'Sindicalizado':
            return $this->calcularVacacionesSindicalizado($aniosTrabajados);
        default:
            return 0;
    }
}
```

//CANTIDAD DE VACACIONES QUE SE CALCULAN PARA EL TIPO DE CONTRATO DE EVENTUAL

```
private function calcularVacacionesEventual($anios)
{
    if ($anios < 1) {
        return 0;
    } elseif ($anios == 1) {
        return 12;
    } elseif ($anios == 2) {
        return 14;
    } elseif ($anios == 3) {
        return 16;
    } elseif ($anios == 4) {
        return 18;
    }
}
```

```
    } elseif ($anios == 5) {
        return 20;
    } elseif ($anios <= 10) {
        return 22;
    } elseif ($anios <= 15) {
        return 24;
    } else {
        return 24 + floor(($anios - 15) / 5) * 2;
    }
}

//CANTIDAD DE VACACIONES QUE SE CALCULAN PARA EL TIPO DE
//CONTRATO DE CONFIANZA
private function calcularVacacionesConfianza($anios)
{
    if ($anios < 1) {
        return 0;
    } elseif ($anios <= 10) {
        return 20;
    } elseif ($anios <= 15) {
        return 22;
    } else {
        return 22 + floor(($anios - 10) / 5) * 2;
    }
}

//CANTIDAD DE VACACIONES QUE SE CALCULAN PARA EL TIPO DE
//CONTRATO SINDICALIZADO
private function calcularVacacionesSindicalizado($anios)
{
    if ($anios < 1) {
        return 0;
    } elseif ($anios <= 10) {
        return 22;
    } elseif ($anios <= 15) {
        return 24;
    } else {
        return 24 + floor(($anios - 10) / 5) * 2;
    }
}
```

- Actualizar y ver datos del empleado.

Descripción. Estas son algunas de las funciones que permiten actualizar los datos del empleado, como se encuentra relacionado con múltiples registros de diferentes tablas, se crearon funciones separadas para llevar un mejor orden. Además, se proporciona la función principal que muestra toda esta información en la vista de empleado/view. En esta función, se generan los registros que se asocian al registro del expediente médico del empleado.

```
//FUNCION PARA CAMBIAR LA FOTO DE PERFIL DEL EMPLEADO
public function actionCambio($id)
{
    $model = $this->findModel2($id);

    if (Yii::$app->request->isPost) {
        $uploadedFile = UploadedFile::getInstance($model, 'foto');
        //RECIBE LA FOTO DEL EMPLEADO, Y SE ALMACENA EN EL
        DIRECTORIO CORRESPONDIENTE
        if ($uploadedFile && $model->validate(['foto'])) {
            $nombreCarpetaTrabajador = Yii::getAlias('@runtime') .
            '/empleados/' . $model->nombre . '_' . $model->apellido;
            if (!is_dir($nombreCarpetaTrabajador)) {
                mkdir($nombreCarpetaTrabajador, 0775, true);
            }

            $nombreCarpetaUsuarioProfile = $nombreCarpetaTrabajador .
            '/foto_empleado';
            if (!is_dir($nombreCarpetaUsuarioProfile)) {
                mkdir($nombreCarpetaUsuarioProfile, 0775, true);
            }

            if ($model->foto && file_exists(Yii::getAlias('@runtime') .
            $model->foto)) {
                unlink(Yii::getAlias('@runtime') . $model->foto);
            }

            $rutaFoto = $nombreCarpetaUsuarioProfile . '/' .
            $uploadedFile->baseName . '.' . $uploadedFile->extension;
            if ($uploadedFile->saveAs($rutaFoto)) {
                $model->foto = $rutaFoto;
                if ($model->save(false)) {

```

```
        Yii::$app->session->setFlash('success', 'La foto del trabajador  
se actualizó correctamente.');//  
    } else {  
        Yii::$app->session->setFlash('error', 'Error al guardar la foto: ' .  
        json_encode($model->errors));  
    }  
} else {  
    Yii::$app->session->setFlash('error', 'Error al guardar la  
imagen.');//  
}  
}  
  
    } else {  
        Yii::$app->session->setFlash('error', 'Error al validar la imagen: ' .  
        json_encode($model->errors));  
    }  
}  
  
return $this->redirect(['view', 'id' => $id]);  
}
```

### //FUNCION QUE PERMITE EDITAR Y ACTUALZAR LA INFORMACION PERSONAL DEL EMPLEADO

```
public function actionActualizarInformacion($id)  
{  
    $model = $this->findModel2($id);  
  
    if ($model->load(Yii::$app->request->post())) {  
        $fechaNacimiento = new \DateTime($model->fecha_nacimiento);  
        $hoy = new \DateTime();  
        $diferencia = $hoy->diff($fechaNacimiento);  
        $model->edad = $diferencia->y;  
  
        if ($model->save()) {  
            Yii::$app->session->setFlash('success', '<i class="fa fa-bell"  
aria-hidden="true"></i>  
Los cambios de información personal han sido actualizados correctamente.');//  
  
            $url = Url::to(['view', 'id' => $model->id,] . '#informacion_personal';  
            return $this->redirect($url);  
        } else {  
            Yii::$app->session->setFlash('error', 'Error al actualizar la información personal');//  
        }  
    }  
}
```

```
Yii::$app->session->setFlash('error', '<i class="fa fa-exclamation-triangle" aria-hidden="true"></i> No se pudieron realizar los cambios de información personal .');
    $url = Url::to(['view', 'id' => $model->id,]) . '#informacion_personal';
    return $this->redirect($url);
}

}

return $this->render('update', [
    'model' => $model,
]);
}

public function actionActualizarInformacionContacto($id)
{
    $model = $this->findModel2($id);

    if ($model->load(Yii::$app->request->post()) && $model->save()) {
        // Yii::$app->session->setFlash('success', 'La información del trabajador ha sido actualizada correctamente.');
        Yii::$app->session->setFlash('success', '<i class="fa fa-bell" aria-hidden="true"></i>
Los cambios de información de contacto han sido actualizados correctamente.');

        $url = Url::to(['view', 'id' => $model->id,]) . '#informacion_contacto';
        return $this->redirect($url);
    } else {
        Yii::$app->session->setFlash('error', '<i class="fa fa-exclamation-triangle" aria-hidden="true"></i> No se pudieron realizar los cambios de información de contacto');
        $url = Url::to(['view', 'id' => $model->id,]) . '#informacion_contacto';
        return $this->redirect($url);
    }
}

//FUNCION QUE PERMITE EDITAR Y ACTUALZAR LA INFORMACION LABORAL DEL EMPLEADO
public function actionActualizarInformacionLaboral($id)
```

```
{  
    $model = $this->findModel2($id);  
    $informacion_laboral =  
    InformacionLaboral::findOne($model->informacion_laboral_id);  
  
    if ($informacion_laboral->load(Yii::$app->request->post())) {  
        // Obtener el nuevo departamento  
        $departamento =  
        CatDepartamento::findOne($informacion_laboral->cat_departamento_id);  
  
        if ($departamento) {  
            // Asignar la dirección y el departamento correspondientes al  
            // departamento seleccionado  
            $informacion_laboral->cat_direccion_id =  
            $departamento->cat_direccion_id;  
            $informacion_laboral->cat_dpto_cargo_id =  
            $departamento->cat_dpto_id;  
        }  
  
        // Guardar los cambios en InformacionLaboral  
        if ($informacion_laboral->save()) {  
            // Actualizar los días de vacaciones  
            $vacaciones =  
            Vacaciones::findOne($informacion_laboral->vacaciones_id);  
            $totalDiasVacaciones =  
            $this->calcularDiasVacaciones($informacion_laboral->fecha_ingreso,  
            $informacion_laboral->cat_tipo_contrato_id);  
            $vacaciones->total_dias_vacaciones = $totalDiasVacaciones;  
  
            if (!$vacaciones->save()) {  
                Yii::$app->session->setFlash('error', 'Hubo un error al actualizar  
                los días de vacaciones.');            }
        }  
  
        // Manejar el cambio en el puesto  
        $nuevoPuesto =  
        CatPuesto::findOne($informacion_laboral->cat_puesto_id);  
        $juntaGobiernoModel = JuntaGobierno::findOne(['empleado_id' =>  
        $model->id, 'cat_direccion_id' => $informacion_laboral->cat_direccion_id]);  
  
        // Verifica si el puesto requiere estar en junta_gobierno  
        if ($nuevoPuesto) {
```

```
$nuevoNivelJerarquico =
$this->determinarNivelJerarquico($nuevoPuesto->nombre_puesto);

if ($nuevoNivelJerarquico !== 'Comun') {
    // Si no existe en junta_gobierno, se agrega
    if (!$juntaGobiernoModel) {
        $juntaGobiernoModel = new JuntaGobierno();
        $juntaGobiernoModel->empleado_id = $model->id;
        $juntaGobiernoModel->nivel_jerarquico =
$nuevoNivelJerarquico;
        $juntaGobiernoModel->cat_departamento_id =
$informacion_laboral->cat_departamento_id;
        $juntaGobiernoModel->cat_direccion_id =
$informacion_laboral->cat_direccion_id;
        if (!$juntaGobiernoModel->save()) {
            Yii::$app->session->setFlash('error', 'Error al guardar
JuntaGobierno.');
        }
    }
} else {
    // Si el puesto no requiere estar en junta_gobierno, eliminar del
registro
    if ($juntaGobiernoModel) {
        $juntaGobiernoModel->delete();
    }
}
}

Yii::$app->session->setFlash('success', 'Los cambios de la
información laboral han sido actualizados correctamente.');
} else {
    Yii::$app->session->setFlash('error', 'Hubo un error al actualizar la
información laboral del trabajador.');
}

$url = Url::to(['view', 'id' => $model->id]) . '#informacion_laboral';
return $this->redirect($url);
} else {
    Yii::$app->session->setFlash('error', 'Hubo un error al cargar los datos
de la solicitud.');
    $url = Url::to(['view', 'id' => $model->id]) . '#informacion_laboral';
    return $this->redirect($url);
```

```
        }
    }

//ESTA FUNCION PERMITE DETERINAR SI ES JEFE DE UNIDAD,
DEPARTAMENTO O SI ES DIRECTOR, EN BASE AL NOMBRE DEL
PUESTO QUE TIENE ESTABLECIDO EL EMPLEADO
protected function determinarNivelJerarquico($nombrePuesto)
{
    if (stripos($nombrePuesto, 'DIRECTOR') !== false) {
        return 'Director';
    } elseif (stripos($nombrePuesto, 'JEFE DE UNIDAD') !== false) {
        return 'Jefe de unidad';
    } elseif (stripos($nombrePuesto, 'JEFE DE DEPARTAMENTO') !== false) {
        return 'Jefe de departamento';
    }
    return 'Comun';
}
```

```
//FUNCION QUE PERMITE EDITAR Y ACTUALZAR LA INFORMACION DE
LOS PERIODOS VACACIONALES DEL EMPLEADO
public function actionActualizarPrimerPeriodo($id)
{
    $model = $this->findModel2($id);
    $periodoVacacional =
$model->informacionLaboral->vacaciones->periodoVacacional;

    if ($periodoVacacional->load(Yii::$app->request->post())) {
        if ($periodoVacacional->dateRange) {
            list($fechalinicio, $fechaFin) = explode(' a ',
$periodoVacacional->dateRange);
            $fechalinicio = new \DateTime($fechalinicio);
            $fechaFin = new \DateTime($fechaFin);
            $diasSeleccionados = $fechalinicio->diff($fechaFin)->days + 1;

            $diasDisponibles = $periodoVacacional->dias_vacaciones_periodo
- $diasSeleccionados;
            $periodoVacacional->dias_disponibles = $diasDisponibles;
        }
    }

    if ($periodoVacacional->save()) {
```

```
    Yii::$app->session->setFlash('success', 'El primer periodo ha sido  
actualizado correctamente.');//  
} else {  
    Yii::$app->session->setFlash('error', 'Hubo un error al actualizar la  
información de vacaciones.');//  
}  
  
$url = Url::to(['view', 'id' => $model->id]) . '#informacion_vacaciones';  
return $this->redirect($url);  
}  
  
$url = Url::to(['view', 'id' => $model->id]) . '#informacion_vacaciones';  
return $this->redirect($url);  
}
```

```
public function actionView($id)  
{  
    //ENCUENTRA EL REGISTRO DEL EMPLEADO SOLICITADO  
    $modelEmpleado = $this->findModel2($id);  
    $modelEmpleado->scenario = Empleado::SCENARIO_UPDATE;  
  
    //CARGA LOS MODELOS  
    $documentos = $modelEmpleado->documentos;  
    $documentoModel = new Documento();  
    $documentoMedicoModel = new DocumentoMedico();  
    $historial = PeriodoVacacionalHistorial::find()->where(['empleado_id' =>  
    $modelEmpleado->id])->all();  
  
    $searchModel = new \app\models\SolicitudSearch();  
    $dataProvider = $searchModel->search(Yii::$app->request->queryParams);  
    $dataProvider->query->andWhere(['empleado_id' => $id]);  
  
    $searchModelConsultas = new \app\models\ConsultaMedicaSearch();  
    $dataProviderConsultas =  
    $searchModelConsultas->search(Yii::$app->request->queryParams);  
    $dataProviderConsultas->query->andWhere(['expediente_medico_id' =>  
    $modelEmpleado->expedienteMedico->id]);  
  
    //CREA EL REGISTRO DE ANTECEDENTE HEREDITARIO Y LO ASOCIA AL  
    EXPEDIENTE MEDICO DEL EMPLEADO
```

```
$expedienteMedico = $modelEmpleado->expedienteMedico;
$antecedenteHereditario =
AntecedenteHereditario::findOne(['expediente_medico_id' =>
$expedienteMedico->id]);
if (!$antecedenteHereditario) {
    $antecedenteHereditario = new AntecedenteHereditario();
    $antecedenteHereditario->expediente_medico_id = $expedienteMedico->id;
    if (!$antecedenteHereditario->save()) {
        Yii::$app->session->setFlash('error', 'Hubo un error al crear el registro de
antecedente hereditario');
    } else {

        $expedienteMedico->antecedente_hereditario_id =
$antecedenteHereditario->id;
        $expedienteMedico->save(false); // Guardar sin validaciones adicionales
    }
}

$antecedentes =
AntecedenteHereditario::find()->where(['expediente_medico_id' =>
$expedienteMedico->id])->all();
$catAntecedentes = CatAntecedenteHereditario::find()->all();

// Obtener o crear el registro de antecedentes patológicos
$antecedentePatologico =
AntecedentePatologico::findOne(['expediente_medico_id' =>
$expedienteMedico->id]);
if (!$antecedentePatologico) {
    $antecedentePatologico = new AntecedentePatologico();
    $antecedentePatologico->expediente_medico_id = $expedienteMedico->id;
    if (!$antecedentePatologico->save()) {
        Yii::$app->session->setFlash('error', 'Hubo un error al crear el registro de
antecedentes patológicos.');
    } else {

        $expedienteMedico->antecedente_patologico_id =
$antecedentePatologico->id;
        $expedienteMedico->save(false); // Guardar sin validaciones adicionales
    }
}
// Obtener o crear el registro de antecedentes no patológicos
```

```
$antecedenteNoPatologico =
AntecedenteNoPatologico::findOne(['expediente_medico_id' =>
$expedienteMedico->id]);
if (!$antecedenteNoPatologico) {
    $antecedenteNoPatologico = new AntecedenteNoPatologico();
    $antecedenteNoPatologico->expediente_medico_id =
$expedienteMedico->id;
    if (!$antecedenteNoPatologico->save()) {
        Yii::$app->session->setFlash('error', 'Hubo un error al crear el registro de
antecedentes no patológicos.');
    } else {

        $expedienteMedico->antecedente_no_patologico_id =
$antecedenteNoPatologico->id;
        $expedienteMedico->save(false); // Guardar sin validaciones adicionales
    }
}

$exploracionFisica = ExploracionFisica::findOne(['expediente_medico_id' =>
$expedienteMedico->id]);
if (!$exploracionFisica) {
    $exploracionFisica = new ExploracionFisica();
    $exploracionFisica->expediente_medico_id = $expedienteMedico->id;
    if (!$exploracionFisica->save()) {
        Yii::$app->session->setFlash('error', 'Hubo un error al crear el registro de
antecedentes no patológicos.');
    } else {

        $expedienteMedico->exploracion_fisica_id = $exploracionFisica->id;
        $expedienteMedico->save(false); // Guardar sin validaciones adicionales
    }
}

$interrogatorioMedico = InterrogatorioMedico::findOne(['expediente_medico_id' =>
$expedienteMedico->id]);
if (!$interrogatorioMedico) {
    $interrogatorioMedico = new InterrogatorioMedico();
    $interrogatorioMedico->expediente_medico_id = $expedienteMedico->id;
    if (!$interrogatorioMedico->save()) {
```

```
Yii::$app->session->setFlash('error', 'Hubo un error al crear el registro de
antecedentes no patológicos.');
} else {

    $expedienteMedico->interrogatorio_medico_id =
$interrogatorioMedico->id;
    $expedienteMedico->save(false); // Guardar sin validaciones adicionales
}
}

$antecedentePerinatal =
AntecedentePerinatal::findOne(['expediente_medico_id' => $expedienteMedico->id]);
if (!$antecedentePerinatal) {
    $antecedentePerinatal = new AntecedentePerinatal();
    $antecedentePerinatal->expediente_medico_id = $expedienteMedico->id;
    if (!$antecedentePerinatal->save()) {
        Yii::$app->session->setFlash('error', 'Hubo un error al crear el registro de
antecedentes no patológicos.');
    } else {

        $expedienteMedico->antecedente_perinatal_id =
$antecedentePerinatal->id;
        $expedienteMedico->save(false); // Guardar sin validaciones adicionales
    }
}

$antecedenteGinecologico =
AntecedenteGinecologico::findOne(['expediente_medico_id' =>
$expedienteMedico->id]);
if (!$antecedenteGinecologico) {
    $antecedenteGinecologico = new AntecedenteGinecologico();
    $antecedenteGinecologico->expediente_medico_id =
$expedienteMedico->id;
    if (!$antecedenteGinecologico->save()) {
        Yii::$app->session->setFlash('error', 'Hubo un error al crear el registro de
antecedentes ginecologicos.');
    } else {
```

```
$expedienteMedico->antecedente_ginecologico_id =
$antecedenteGinecologico->id;
    $expedienteMedico->save(false); // Guardar sin validaciones adicionales
}
}

$antecedenteObstrectico =
AntecedenteObstrectico::findOne(['expediente_medico_id' =>
$expedienteMedico->id]);
if (!$antecedenteObstrectico) {
    $antecedenteObstrectico = new AntecedenteObstrectico();
    $antecedenteObstrectico->expediente_medico_id = $expedienteMedico->id;
    if (!$antecedenteObstrectico->save()) {
        Yii::$app->session->setFlash('error', 'Hubo un error al crear el registro de
antecedentes obstrecticos.');
    } else {

        $expedienteMedico->antecedente_obstrectico_id =
$antecedenteObstrectico->id;
        $expedienteMedico->save(false); // Guardar sin validaciones adicionales
    }
}

$alergia = Alergia::findOne(['expediente_medico_id' =>
$expedienteMedico->id]);
if (!$alergia) {
    $alergia = new Alergia();
    $alergia->expediente_medico_id = $expedienteMedico->id;
    if (!$alergia->save()) {
        Yii::$app->session->setFlash('error', 'Hubo un error al crear el registro de
alergia.');
    } else {

        $expedienteMedico->alergia_id = $alergia->id;
        $expedienteMedico->save(false); // Guardar sin validaciones adicionales
    }
}

$expedienteMedico = $modelEmpleado->expedienteMedico;
```

```
if ($expedienteMedico && !$expedienteMedico->empleado_id) {
    $expedienteMedico->empleado_id = $modelEmpleado->id;
    if ($expedienteMedico->save(false)) {
        // Yii::$app->session->setFlash('success', 'Empleado asignado al
expediente médico.');
    } else {
        Yii::$app->session->setFlash('error', 'Hubo un error al asignar el empleado
al expediente médico.');
    }
}

if (Yii::$app->request->isPost) {
    if ($post = Yii::$app->request->post('AntecedenteHereditario')) {
        $observacionGeneral = Yii::$app->request->post('observacion_general');

        // AntecedenteHereditario::deleteAll(['expediente_medico_id' =>
$expedienteMedico->id]);

        foreach ($post as $catAntecedentId => $parentezcos) {
            foreach ($parentezcos as $parentezco => $value) {
                $antecedente = new AntecedenteHereditario();
                $antecedente->expediente_medico_id = $expedienteMedico->id;
                $antecedente->cat_antecedente_hereditario_id = $catAntecedentId;
                $antecedente->parentezco = $parentezco;
                $antecedente->observacion = $observacionGeneral;
                $antecedente->save();
            }
        }
    }

    Yii::$app->session->setFlash('success', 'Información de antecedentes
hereditarios guardada correctamente.');
    $url = Url::to(['view', 'id' => $id]) . '#hereditarios';
    return $this->redirect($url);
}

if ($descripcionAntecedentes =
Yii::$app->request->post('descripcion_antecedentes')) {
    $antecedentePatologico->descripcion_antecedentes =
$descripcionAntecedentes;
```

```
if ($antecedentePatologico->save()) {
    Yii::$app->session->setFlash('success', 'Información de antecedentes
    patológicos guardada correctamente.');
} else {
    Yii::$app->session->setFlash('error', 'Hubo un error al guardar la
    información de antecedentes patológicos.');
}

$url = Url::to(['view', 'id' => $id]) . '#expediente_medico';
return $this->redirect($url);
}

return $this->render('view', [
    'model' => $modelEmpleado,
    'documentos' => $documentos,
    'documentoModel' => $documentoModel,
    'documentoMedicoModel' => $documentoMedicoModel,
    'historial' => $historial,
    'searchModelConsultas' => $searchModelConsultas,
    'dataProviderConsultas' => $dataProviderConsultas,
    'searchModel' => $searchModel,
    'dataProvider' => $dataProvider,

    'expedienteMedico' => $expedienteMedico,
    'antecedentes' => $antecedentes,
    'catAntecedentes' => $catAntecedentes,
    'antecedenteNoPatologico' => $antecedenteNoPatologico,
    'ExploracionFisica' => $exploracionFisica,
    'InterrogatorioMedico' => $interrogatorioMedico,
    'AntecedentePerinatal' => $antecedentePerinatal,
    'AntecedenteGinecologico' => $antecedenteGinecologico,
    'AntecedenteObstrectico' => $antecedenteObstrectico,
    'Alergia' => $alergia,
    'antecedentePatologico' => $antecedentePatologico,
]);

}
```

- Crear documentos, subir, descargar y visualizar archivos.

Descripción. Código que permite crear los registros de los documentos que se asocian al empleado (los documentos médicos siguen una lógica muy similar). Aquí se muestra como se crea el registro del documento y a su vez este se sube al directorio que pertenece al empleado en la carpeta de runtime, de esa forma se almacenan los archivos. Además, las funciones de download y open, se encargan de identificar el registro del archivo para su posible visualización o descarga de este.

```
public function actionCreate($empleado_id)
{
    //CARGA LOS MODELOS
    $model = new Documento();
    $cat_tipo_documento = new CatTipoDocumento();

    $model->empleado_id = $empleado_id;

    if (Yii::$app->request->isPost) {
        $model->load(Yii::$app->request->post());
        $file = UploadedFile::getInstance($model, 'ruta');
        $tipoDocumentoidSeleccionado =
            Yii::$app->request->post('Documento')['cat_tipo_documento_id'];
        $nombreDocumento = Yii::$app->request->post('Documento')['nombre'];

        // Verificar si se seleccionó 'OTRO' y se ingresó un nuevo nombre de
        documento
        if ($tipoDocumentoidSeleccionado == 4 && !empty($nombreDocumento)) {
            $nuevoTipoDocumento = new CatTipoDocumento();
            $nuevoTipoDocumento->nombre_tipo = $nombreDocumento;
            if ($nuevoTipoDocumento->save()) {
                // Crear el nuevo tipo de documento y obtener su ID
                $tipoDocumentoidSeleccionado = $nuevoTipoDocumento->id;
                Yii::$app->session->setFlash('success', 'Nuevo tipo de documento
creado exitosamente.');
            } else {
                Yii::$app->session->setFlash('error', 'No se pudo crear el nuevo tipo de
documento.');
            }
            return $this->render('create', [
                'model' => $model,
                'cat_tipo_documento' => $cat_tipo_documento,
            ]);
        }
    }

    $model->cat_tipo_documento_id = $tipoDocumentoidSeleccionado;
```

```
// Asignar el nombre del documento solo si es 'OTRO'
if ($tipoDocumentIdSeleccionado != 4) {
    $nombreDocumento =
CatTipoDocumento::findOne($tipoDocumentIdSeleccionado)->nombre_tipo;
}
$model->nombre = $nombreDocumento;
//LOGICA DE DIRECTORIOS
if ($file && $model->cat_tipo_documento_id !== null) {
    $empleado = Empleado::findOne($model->empleado_id);
    //BUSCA EL DIRECTORIO DEL EMPLEADO
    $nombreCarpetaTrabajador = Yii::getAlias('@runtime') . '/empleados/' .
$empleado->nombre . '_' . $empleado->apellido;
    if (!is_dir($nombreCarpetaTrabajador)) {
        mkdir($nombreCarpetaTrabajador, 0775, true);
    }
    //CREA LA CARPETA DE DOCUMENTOS EN EL DIRECTORIO
    $nombreCarpetaExpedientes = $nombreCarpetaTrabajador .
'/documentos';
    if (!is_dir($nombreCarpetaExpedientes)) {
        mkdir($nombreCarpetaExpedientes, 0775, true);
    }
    //GUARDA EL ARCHIVO EN EL DIRECTORIO
    $rutaArchivo = $nombreCarpetaExpedientes . '/' . $file->baseName . '.' .
$file->extension;
    $file->saveAs($rutaArchivo);

$model->ruta = $rutaArchivo;
$model->fecha_subida = date('Y-m-d H:i:s');

if ($model->save()) {
    Yii::$app->session->setFlash('success', 'El documento se ha subido
exitosamente.');
    $url = Url::to(['empleado/view', 'id' => $empleado->id]) . '#documentos';
    return $this->redirect($url);
}
}

return $this->render('create', [
    //RENDERIZA LA VISTA Y CARGA LOS MODELOS CORRESPONDIENTES
    'model' => $model,
    'cat_tipo_documento' => $cat_tipo_documento,
]);
}

//ABRIR DOCUMENTO
```

```
public function actionOpen($id)
{
    //EN CASO DE QUE SE TRATE DE UN TIPO DE DOCUMENTO QUE SE PUEDA
    ABRIR O VISUALIZAR
    //DESDE EL NAVEGADOR, ESTA FUNCION SE ENCARGAR
    $model = Documento::findOne($id);
    if ($model) {
        $rutaArchivo = $model->ruta;
        //BUSCA LA RUTA DEL ARCHIVO
        if (file_exists($rutaArchivo)) {
            return Yii::$app->response->sendFile($rutaArchivo, $model->nombre, ['inline'=> true]);
        }
    }
    Yii::$app->session->setFlash('error', 'El archivo solicitado no se encuentra
disponible.');
    return $this->redirect(['empleado/index']);
}
//DESCARGAR DOCUMENTO
public function actionDownload($id)
{
    //SI SE DESEA DESCARGAR EL DOCUMENTO, ESTA FUNCION
    //BUSCA EL REGISTRO DEL DOCUMENTO, PARA QUE USANDO LA RUTA
    //ENCONTRAR Y OBTENER EL ARCHIVO, ASI PODERLO DESCARGAR
    $model = Documento::findOne($id);
    if ($model) {
        $rutaArchivo = $model->ruta;
        if (file_exists($rutaArchivo)) {
            $nombreArchivo = $model->nombre;
            $extension = pathinfo($rutaArchivo, PATHINFO_EXTENSION);
            $nombreCompleto = $nombreArchivo . '.' . $extension;
            return Yii::$app->response->sendFile($rutaArchivo, $nombreCompleto);
        }
    }
    Yii::$app->session->setFlash('error', 'El archivo solicitado no se encuentra
disponible.');
    return $this->redirect(['documento/index']);
}
```

- Ver solicitudes.

Descripción. Código que permite identificar que tipo de solicitud es la que se quiere visualizar y así mismo cargar los atributos correspondientes al tipo de

formato asociado a la solicitud. En conjunto a la función getFormatoAsociado, identifica el tipo de solicitud de solicitud que se va a mostrar.

```
//MOSTRAR LA SOLICITUD CON EL REGISTRO DEL FORMATO
ASOCIADO
public function actionView($id)
{
    $model = $this->findModel($id);

    if (Yii::$app->user->can('medico') || Yii::$app->user->can('gestor-rh')) {
        if ($model->status === 'Nueva') {
            $model->status = 'Visto';
            $model->save(false); // Guarda el cambio sin validar el modelo
        }
    }

    // Obtener el formato asociado
    $formato = $this->getFormatoAsociado($model);
    $empleadold = $model->empleado_id;

    // Encontrar el ReporteTiempoExtra asociado usando el solicitud_id del
    // modelo
    $reporteTiempoExtra = ReporteTiempoExtra::find()->where(['solicitud_id' => $model->id])->one();

    // Si se encuentra el ReporteTiempoExtra, obtener las actividades
    // asociadas
    $actividades = [];
    if ($reporteTiempoExtra) {
        $actividades =
        ActividadReporteTiempoExtra::find()->where(['reporte_tiempo_extra_id' => $reporteTiempoExtra->id])->all();
    }

    return $this->render('view', [
        'model' => $model,
        'formato' => $formato,
        'empleadold' => $empleadold,
        'actividades' => $actividades, // Pasar actividades a la vista
    ]);
}
```

```
//OBTENER EL FORMATO ASOCIADO A LA SOLICITUD
protected function getFormatoAsociado($solicitud)
{
    //EN BASE A LOS DIFERENTES MODELOS DE FORMATOS QUE
    EXISTEN
    //SE IDENTIFICA LA SOLICITUD ASOCIADA A LOS REGISTROS
    //Y SE SABE DE QUE TIPO DE SOLICITUD ES LA QUE SE ESTA
    SOLICITANDO
    switch ($solicitud->nombre_formato) {
        case 'PERMISO FUERA DEL TRABAJO':
            return PermisoFueraTrabajo::findOne(['solicitud_id' =>
$solicitud->id]);
        case 'CITA MEDICA':
            return CitaMedica::findOne(['solicitud_id' => $solicitud->id]);
        case 'COMISION ESPECIAL':
            return ComisionEspecial::findOne(['solicitud_id' => $solicitud->id]);
        case 'PERMISO ECONOMICO':
            return PermisoEconomico::findOne(['solicitud_id' => $solicitud->id]);
        case 'CAMBIO DE DÍA LABORAL':
            return CambioDiaLaboral::findOne(['solicitud_id' => $solicitud->id]);
        case 'CAMBIO DE HORARIO DE TRABAJO':
            return CambioHorarioTrabajo::findOne(['solicitud_id' =>
$solicitud->id]);
        case 'PERMISO SIN GOCE DE SUELDO':
            return PermisoSinSueldo::findOne(['solicitud_id' => $solicitud->id]);
        case 'CAMBIO DE PERIODO VACACIONAL':
            return CambioPeriodoVacacional::findOne(['solicitud_id' =>
$solicitud->id]);
        case 'REPORTE DE TIEMPO EXTRA':
            return ReporteTiempoExtra::findOne(['solicitud_id' =>
$solicitud->id]);
        default:
            return null;
    }
}
```

- Controlador del sitio.

Descripción. Código que tiene funciones principales del sitio en general, como lo son; el inicio de sesión y el cambio de contraseña en el caso de que sea un usuario nuevo. La configuración de algunos apartados del sistema, como lo son los parámetros de los formatos donde se establece de la cantidad de permisos anuales que tienen permitidos los empleados en base a su tipo de contrato. También se encuentran las funciones que muestran los inicios (home), dependiendo de los roles de los usuarios, como el inicio para médico y el de los empleados (también conocido como “mi perfil”).

//FUNCIÓN QUE SE ENCARGA DE RECIBIR LAS CREDENCIALES INGRESADAS

//POR LOS USUARIOS Y REDIRIGIRLOS A LAS VISTAS CORRESPONDIENTES A SU ROL

```
public function actionLogin()
{
    if (!Yii::$app->user->isGuest) {
        $userId = Yii::$app->user->identity->id;
        $user = Usuario::findOne($userId);

        if ($user->nuevo == 4) {
            return $this->redirect(['usuario/cambiarcontrasena']);
        }
        elseif (Usuario::isUserAdmin($userId)) {
            return $this->redirect(["site/portalgestionrh"]);
        } elseif(Usuario:: isUserSimple($userId)) {
            return $this->redirect(["site/portalempelado"]);
        }else{
            return $this->redirect(["site/portal-medico"]);
        }
    }

    $model = new LoginForm();
    if ($model->load(Yii::$app->request->post()) && $model->login()) {
        $userId = Yii::$app->user->identity->id;
        $user = Usuario::findOne($userId);
        $userId = Yii::$app->user->identity->id;
        if ($user->nuevo == 4) {
            return $this->redirect(["usuario/cambiarcontrasena"]);
        } else {
            if (Usuario::isUserAdmin($userId)) {
```

```
        return $this->redirect(['site/portalgestionrh']);
}elseif (Usuario::isUserSimple($userId)) {

    return $this->redirect(["site/portalepleado"]);
}
else{

    return $this->redirect(["site/portal-medico"]);

}

}

} else {
    $model->password = "";
    return $this->render('login', [
        'model' => $model,
    ]);
}
}

// FUNCION QUE SE ENCARGA DE RENDERIZAR LA VISTA DE
CAMBIAR CONTRASEÑA
// ESTA FUNCION RECIBE LOS DATOS DEL FORMULARIO PARA EL
CAMBIO DE CONTRASEÑA
// DE LOS USUARIOS NUEVOS
public function actionCambiarcontrasena()
{
    $model = new CambiarContrasena();

    if ($model->load(Yii::$app->request->post()) &&
$model->changePassword()) {
        $userId = Yii::$app->user->identity->id;
        $user = Usuario::findOne($userId);

        if (Usuario::isUserAdmin($userId)) {
            return $this->redirect(['site/portalgestionrh']);
        }
        elseif (Usuario::isUserMedico($userId)) {
            return $this->redirect(['site/portal-medico']);
        }
    }
}
```

```
}

elseif (Usuario:::isUserSimple($userId)) {
    return $this->redirect(['site/portalempleado']);
}
}

return $this->render('cambiarcontrasena', [
    'model' => $model,
]);
}

//SE RENDERIZA LA VISTA DE CONFIGURACIONES DONDE SE
CARGAN LOS REGISTROS DE LOS MODELOS
//DE CAT_DEPARTAMENTO, PARAMETRO_FORMATO, CAT_PUESTO
public function actionConfiguracion()
{
    $parametroFormatosearchModel = new ParametroFormatoSearch();
    $parametroFormatoDataProvider =
$parametroFormatosearchModel->search(Yii::$app->request->queryParams);

    $catPuestosearchModel = new CatPuestoSearch();
    $catPuestoDataProvider =
$catPuestosearchModel->search(Yii::$app->request->queryParams);

    $catDepartamentosearchModel = new CatDepartamentoSearch();
    $catDepartamentoDataProvider =
$catDepartamentosearchModel->search(Yii::$app->request->queryParams);

    return $this->render('configuracion', [
        'parametroFormatosearchModel' =>
$parametroFormatosearchModel,
        'parametroFormatoDataProvider' => $parametroFormatoDataProvider,
        'catPuestosearchModel' => $catPuestosearchModel,
        'catPuestoDataProvider' => $catPuestoDataProvider,
        'catDepartamentosearchModel' => $catDepartamentosearchModel,
        'catDepartamentoDataProvider' => $catDepartamentoDataProvider
    ]);
}
```

```
    ]);
}

/**
 * Logout action.
 *
 * @return Response
 */
//FUNCION QUE SE ENCARGA DE CERRAR LA SESIÓN DE LOS
USUARIOS
public function actionLogout()
{
    Yii::$app->user->logout();
    return $this->redirect(['site/login']);
}

/**
 * Displays contact page.
 *
 * @return Response|string
 */
/**
 * Displays about page.
 *
 * @return string
 */

// FUNCION QUE SE ENCARGA DE RENDERIZAR EL INICIO DEL
USUARIO MEDICO
// BUSCA LAS SOLICITUDES DE CITAS MEDICAS MÁS RECIENTES
public function actionPortalMedico()
{
    $solicitudesRecientes = \app\models\Solicitud::find()
        ->where(['nombre_formato' => 'CITA MEDICA'])
        ->orderBy(['fecha_creacion' => SORT_DESC])
        ->limit(10)
        ->all();
```

```
$usuarioid = Yii::$app->user->identity->id;
$notificacionesCount = NotificacionMedico::countUnread($usuarioid);

return $this->render('portal-medico', [
    'solicitudesRecientes' => $solicitudesRecientes,
    'notificacionesCount' => $notificacionesCount,
]);
}

// FUNCION QUE SE ENCARGA DE OBTENER LA FOTO DEL USUARIO
// LA OBTIENE DE LA CARPETA DEL EMPLEADO QUE SE ENCUENTRA
// EN EL DIRECTORIO DE RUNTIME
public function actionGetEmpleadoFoto($filename)
{
    $filePath = Yii::getAlias('@runtime') . '/empleados/' . $filename;
    if (file_exists($filePath)) {
        return Yii::$app->response->sendFile($filePath);
    } else {
        throw new \yii\web\NotFoundHttpException("File not found");
    }
}

// FUNCION QUE RENDERIZA EL INICIO DEL USUARIO EMPLEADO
// SE CARGA LA INFORMACIÓN ESPECIFICA DEL EMPLEADO EN
BASE A LOS MODELOS
// PROPORCIONADOS
public function actionPortalemppleado()
{

$avisos = \app\models\Aviso::find()->all();
```

```
$usuario = Yii::$app->user->identity;
$modelEmpleado = $usuario->empleado;

if (!$modelEmpleado) {
    throw new NotFoundHttpException('El empleado no existe.');
}

$documentos = $modelEmpleado->documentos;
$documentoModel = new Documento();
$documentoMedicoModel = new DocumentoMedico();
$historial = PeriodoVacacionalHistorial::find()->where(['empleado_id' =>
$modelEmpleado->id])->all();

$searchModel = new \app\models\SolicitudSearch();
$dataProvider =
$searchModel->search(Yii::$app->request->queryParams);
    $dataProvider->query->andWhere(['empleado_id' =>
$modelEmpleado->id]);

$searchModelConsultas = new \app\models\ConsultaMedicaSearch();
$dataProviderConsultas =
$searchModelConsultas->search(Yii::$app->request->queryParams);
    $dataProviderConsultas->query->andWhere(['expediente_medico_id' =>
$modelEmpleado->expedienteMedico->id]);

$expedienteMedico = $modelEmpleado->expedienteMedico;
$antecedentes =
AntecedenteHereditario::find()->where(['expediente_medico_id' =>
$expedienteMedico->id])->all();
$catAntecedentes = CatAntecedenteHereditario::find()->all();

$antecedentePatologico =
AntecedentePatologico::findOne(['expediente_medico_id' =>
$expedienteMedico->id]) ?: new
AntecedentePatologico(['expediente_medico_id' => $expedienteMedico->id]);
$antecedenteNoPatologico =
AntecedenteNoPatologico::findOne(['expediente_medico_id' =>
```

```
$expedienteMedico->id]) ?: new
AntecedenteNoPatologico(['expediente_medico_id' =>
$expedienteMedico->id]);
    $ExploracionFisica =
ExploracionFisica::findOne(['expediente_medico_id' =>
$expedienteMedico->id]) ?: new ExploracionFisica(['expediente_medico_id' =>
$expedienteMedico->id]);
    $InterrogatorioMedico =
InterrogatorioMedico::findOne(['expediente_medico_id' =>
$expedienteMedico->id]) ?: new
InterrogatorioMedico(['expediente_medico_id' => $expedienteMedico->id]);
    $AntecedentePerinatal =
AntecedentePerinatal::findOne(['expediente_medico_id' =>
$expedienteMedico->id]) ?: new
AntecedentePerinatal(['expediente_medico_id' => $expedienteMedico->id]);
    $AntecedenteGinecologico =
AntecedenteGinecologico::findOne(['expediente_medico_id' =>
$expedienteMedico->id]) ?: new
AntecedenteGinecologico(['expediente_medico_id' =>
$expedienteMedico->id]);
    $AntecedenteObstrectico =
AntecedenteObstrectico::findOne(['expediente_medico_id' =>
$expedienteMedico->id]) ?: new
AntecedenteObstrectico(['expediente_medico_id' =>
$expedienteMedico->id]);
    $Alergia = Alergia::findOne(['expediente_medico_id' =>
$expedienteMedico->id]) ?: new Alergia(['expediente_medico_id' =>
$expedienteMedico->id]);

return $this->render('portalempleado', [
    'model' => $modelEmpleado,
    'documentos' => $documentos,
    'documentoModel' => $documentoModel,
    'historial' => $historial,
    'searchModelConsultas' => $searchModelConsultas,
    'dataProviderConsultas' => $dataProviderConsultas,
    'searchModel' => $searchModel,
    'documentoMedicoModel' => $documentoMedicoModel,
    'dataProvider' => $dataProvider,
    'expedienteMedico' => $expedienteMedico,
    'antecedentes' => $antecedentes,
    'catAntecedentes' => $catAntecedentes,
```

```
'antecedenteNoPatologico' => $antecedenteNoPatologico,  
'ExploracionFisica' => $ExploracionFisica,  
'InterrogatorioMedico' => $InterrogatorioMedico,  
'AntecedentePerinatal' => $AntecedentePerinatal,  
'AntecedenteGinecologico' => $AntecedenteGinecologico,  
'AntecedenteObstrectico' => $AntecedenteObstrectico,  
'Alergia' => $Alergia,  
'antecedentePatologico' => $antecedentePatologico,  
'avisos' => $avisos,  
]);  
}
```

### - Controladores de los formatos.

La creación de los formatos siguen un mismo tipo de logica y estructura con variaciones minimas. Todos tienen estas funciones principales que son con las que trabaja el sistema: actionHistorial, actionView, actionIndex, actionCreate, actionDelete, actionExportHtml y actionExportHtmlSegundoCaso.

- Descripción.
  - Historial e Index se encargan de obtener todos los registros creados de ese formato por el empleado.
  - View obtiene la información detallada del registro del formato que creó el empleado.
  - Create genera el registro del formato, como se encuentra asociado a solicitud, también crea el registro de solicitud.
  - ExportHtml y ExportHtmlSegundoCaso son las funciones encargadas de leer la plantilla del formato (que se encuentra como un archivo excel), después obtiene la información del registro del formato creado y escribe en las celdas correspondientes del excel, para su posterior conversión a Html y este pueda ser impreso. El segundo caso sigue la misma lógica, solo que este se utiliza para los casos especiales de empleados, y solo varía el cómo se escribe la información en la plantilla.
  - Delete es la función encargada de eliminar el registro del formato, pero a su vez borra también el registro de la solicitud asociada a este.

### ● historial e index.

```
//FUNCION PARA MOSTRAR TODOS LO REGISTROS DE ESTE FORMATO  
CREADOS POR EL EMPLEADO  
public function actionIndex()
```

```
{  
    //OBTIENE LOS REGISTROS SOLO DEL USUARIO IDENTIFICADO  
    $usuarioid = Yii::$app->user->identity->id;  
  
    $empleado = Empleado::find()->where(['usuario_id' => $usuarioid])->one();  
  
    if ($empleado !== null) {  
        $searchModel = new PermisofueraTrabajoSearch();  
        $dataProvider = $searchModel->search(Yii::$app->request->queryParams);  
  
        $dataProvider->query->andFilterWhere(['empleado_id' => $empleado->id]);  
  
        return $this->render('index', [  
            'searchModel' => $searchModel,  
            'dataProvider' => $dataProvider,  
        ]);  
    } else {  
        Yii::$app->session->setFlash('error', 'No se pudo encontrar el empleado  
asociado al usuario actual.');//  
        return $this->redirect(['index']);  
    }  
}//FUNCION PARA MOSTRAR TODOS LO REGISTROS DE ESTE FORMATO  
CREADOS POR EL EMPLEADO  
  
public function actionHistorial($empleado_id= null)  
{  
    $empleado = Empleado::findOne($empleado_id);  
  
    if ($empleado === null) {  
        throw new NotFoundHttpException('El empleado seleccionado no existe.');//  
    }  
  
    $searchModel = new PermisofueraTrabajoSearch();  
    $dataProvider = $searchModel->search(Yii::$app->request->queryParams);  
    $dataProvider->query->andFilterWhere(['empleado_id' => $empleado->id]);  
  
    $this->layout = "main-trabajador";  
  
    return $this->render('historial', [  
        'searchModel' => $searchModel,  
        'dataProvider' => $dataProvider,  
        'empleado' => $empleado,  
    ]);  
}
```

- view y delete.

//FUNCIÓN PARA MOSTRAR LA INFORMACIÓN DETALLADA DEL REGISTRO DEL FORMATO

```
public function actionView($id)
{
    $model = $this->findModel($id);
    $empleado = Empleado::findOne($model->empleado_id);
//OBTIENE EL REGISTRO DEL EMPLEADO ASOCIADO
    return $this->render('view', [
        'model' => $model,
        'empleado' => $empleado, // Pasar empleado a la vista
    ]);
}
```

//FUNCIÓN PARA ELIMINAR EL REGISTRO DEL FORMATO Y EL REGISTRO DE SOLICITUD ASOCIADO

```
public function actionDelete($id)
{
    $model = $this->findModel($id);
    $empleado = Empleado::findOne($model->empleado_id);
    if ($model === null) {
        Yii::$app->session->setFlash('error', 'El registro de permiso no fue
encontrado.');
        return $this->redirect(['index']);
    }
}
```

```
$transaction = Yii::$app->db->beginTransaction();
try {
    // Obtener el ID de la solicitud asociada antes de eliminar el permiso
    $solicitudId = $model->solicitud_id;
```

```
    // Eliminar el registro de PermisoFueraTrabajo
    if ($model->delete()) {
        // Buscar y eliminar el registro asociado en la tabla Solicitud
        $solicitudModel = Solicitud::findOne($solicitudId);
        if ($solicitudModel !== null) {
            $solicitudModel->delete();
        }
    }
}
```

```
$transaction->commit();
```

```
    Yii::$app->session->setFlash('success', 'El registro de permiso y la  
solicitud asociada han sido eliminados correctamente.');
```

```
    } else {  
        Yii::$app->session->setFlash('error', 'Hubo un error al eliminar el registro  
de permiso.');
```

```
    }  
} catch (\Exception $e) {  
    $transaction->rollBack();  
    Yii::$app->session->setFlash('error', 'Hubo un error al eliminar el registro: ' .  
$e->getMessage());  
} catch (\Throwable $e) {  
    $transaction->rollBack();  
    Yii::$app->session->setFlash('error', 'Hubo un error al eliminar el registro: ' .  
$e->getMessage());  
}
```

```
if (Yii::$app->user->can('gestor-rh')) {  
    Yii::$app->session->setFlash('success', 'El registro se ha eliminado  
exitosamente.');
```

```
    $url = Url::to(['historial', 'empleado_id' => $empleado->id]);  
    return $this->redirect($url);  
} else {  
    return $this->redirect(['index']);  
}  
}
```

- create.

```
//FUNCIÓN QUE CREA EL REGISTRO DEL FORMATO  
public function actionCreate($empleado_id = null)  
{  
    $this->layout = "main-trabajador";  
    //OBTIENE LOS MODELOS ASOCIADOS  
    $model = new PermisoFueraTrabajo();  
    $motivoFechaPermisoModel = new MotivoFechaPermiso(); //PARA  
    REGISTRAR FECHA Y MOTIVO DEL PERMISO  
  
    $solicitudModel = new Solicitud();  
    $usuarioid = Yii::$app->user->identity->id;
```

```
/VERIFICA SI EL REGISTRO ESTA SIENDO CREADO POR EL MISMO  
EMPLEADO CON LA  
//SESION INICIADA, O SI OTRO EMPLEADO ESTA CREANDO SU REGISTRO  
if ($empleado_id) {  
    $empleado = Empleado::findOne($empleado_id);  
} else {  
    $empleado = Empleado::find()->where(['usuario_id' => $usuariold])->one();  
}  
  
if ($empleado) {  
    $model->empleado_id = $empleado->id;  
} else {  
    Yii::$app->session->setFlash('error', 'No se pudo encontrar el empleado.');//  
    return $this->redirect(['index']);  
}  
  
// Calcular permisos usados y disponibles  
$year = date('Y');  
$tipoContratold = $empleado->informacionLaboral->cat_tipo_contrato_id;  
  
$parametroFormato = ParametroFormato::find()  
    ->where(['tipo_permiso' => 'PERMISO FUERA DEL TRABAJO',  
    'cat_tipo_contrato_id' => $tipoContratold])  
    ->one();  
  
if (!$parametroFormato) {  
    Yii::$app->session->setFlash('error', 'No se pudo encontrar el parámetro de  
formato para tu tipo de contrato.');//  
    return $this->redirect(['index']);  
}  
  
$totalPermisosAnuales = $parametroFormato->limite_anual;  
  
$permisosUsados = PermisoFueraTrabajo::find()  
    ->where(['empleado_id' => $empleado->id])  
    ->andWhere(['between', 'fecha_a_reponer', "$year-01-01", "$year-12-31"])  
    ->count();  
  
$permisosDisponibles = $totalPermisosAnuales - $permisosUsados;  
  
// Verificar si se alcanzó el límite de permisos  
if ($permisosDisponibles <= 0) {  
    Yii::$app->session->setFlash('error', 'Has alcanzado el límite anual de  
permisos.');//  
    // return $this->redirect(['index']);
```

```
}

if ($model->load(Yii::$app->request->post()) &&
$motivoFechaPermisoModel->load(Yii::$app->request->post())) {
    $transaction = Yii::$app->db->beginTransaction();
    try {
        if ($motivoFechaPermisoModel->save()) {
            $model->motivo_fecha_permiso_id = $motivoFechaPermisoModel->id;
            $model->hora_salida = date('H:i:s', strtotime($model->hora_salida));
            $model->hora_Regreso = date('H:i:s', strtotime($model->hora_Regreso));
            $model->horario_fecha_a_reponer = date('H:i:s',
                strtotime($model->horario_fecha_a_reponer));
            //GENERA EL REGISTRO DE LA SOLICITUD Y ASIGNA DATOS AL
            REGISTRO
            $solicitudModel->empleado_id = $empleado->id;
            $solicitudModel->status = 'Nueva';
            $solicitudModel->comentario = '';
            $solicitudModel->fecha_aprobacion = null;
            $solicitudModel->fecha_creacion = date('Y-m-d H:i:s');
            $solicitudModel->nombre_formato = 'PERMISO FUERA DEL
            TRABAJO';
            if ($solicitudModel->save()) {
                $model->solicitud_id = $solicitudModel->id;

                if ($model->jefe_departamento_id) {
                    $jefeDepartamento =
                        JuntaGobierno::findOne($model->jefe_departamento_id);
                    $model->nombre_jefe_departamento = $jefeDepartamento ?
                        $jefeDepartamento->empleado->profesion . ' .
                        $jefeDepartamento->empleado->nombre . ' .
                        $jefeDepartamento->empleado->apellido : null;
                }

                if ($model->save()) {
                    $transaction->commit();
                    Yii::$app->session->setFlash('success', 'Su solicitud ha sido
                    generada exitosamente.');

                    $notificacion = new Notificacion();
                    $notificacion->usuario_id = $model->empleado->usuario_id;
                    $notificacion->mensaje = 'Tienes una nueva solicitud pendiente de
                    revisión.';
                    $notificacion->created_at = date('Y-m-d H:i:s');
                    $notificacion->leido = 0;
                    if ($notificacion->save()) {
                        return $this->redirect(['view', 'id' => $model->id]);
                    }
                }
            }
        }
    }
}
```

```
        } else {
            Yii::$app->session->setFlash('error', 'Hubo un error al guardar la
notificación.');
        }
    } else {
        Yii::$app->session->setFlash('error', 'Hubo un error al guardar la
solicitud.');
    }
}
}
$transaction->rollBack();
} catch (\Exception $e) {
    $transaction->rollBack();
    Yii::$app->session->setFlash('error', 'Hubo un error al crear el registro: ' .
$e->getMessage());
} catch (\Throwable $e) {
    $transaction->rollBack();
    Yii::$app->session->setFlash('error', 'Hubo un error al crear el registro: ' .
$e->getMessage());
}
}

return $this->render('create', [
    'model' => $model,
    'motivoFechaPermisoModel' => $motivoFechaPermisoModel,
    'solicitudModel' => $solicitudModel,
    'empleado' => $empleado,
    'permisosUsados' => $permisosUsados,
    'permisosDisponibles' => $permisosDisponibles,
]);
}
```

- exportHtml y exportHtmlSegundoCaso.

```
//FUNCION QUE LEE LA PLANTILLA EXCEL, ESCRIBE SOBRE ESTA MISMA Y
CONVIERTE A HTML
public function actionExportHtmlSegundoCaso($id)
{
    $this->layout = false;

    $model = PermisoFueraTrabajo::findOne($id);

    if (!$model) {
        throw new NotFoundHttpException('El registro no existe.');
    }
```

```
//LEE LA PLANTILLA EXCEL
$templatePath = Yii::getAlias('@app/templates/permiso_fuera_trabajo.xlsx');

$spreadsheet = IOFactory::load($templatePath);

$sheet = $spreadsheet->getActiveSheet();

//COMIENZA A ESCRIBIR LOS VALORES DEL REGISTRO DEL FORMATO
SOBRE LAS CELDAS ESPECIFICADAS
$sheet->setCellValue('F6', $model->empleado->numero_empleado);

$nombre = mb_strtoupper($model->empleado->nombree, 'UTF-8');
$apellido = mb_strtoupper($model->empleado->apellido, 'UTF-8');
$nombreCompleto = $apellido . ' ' . $nombre;
$sheet->setCellValue('H7', $nombreCompleto);

setlocale(LC_TIME, 'es_419.UTF-8');
$fechaHOY = strftime("%A, %B %d, %Y"); //formato de fecha
$sheet->setCellValue('N6', $fechaHOY);

$nombrePuesto =
$model->empleado->informacionLaboral->catPuesto->nombre_puesto;
$sheet->setCellValue('H8', $nombrePuesto);

$nombreCargo =
$model->empleado->informacionLaboral->catDptoCargo->nombre_dpto;
$sheet->setCellValue('H9', $nombreCargo);

$nombreDireccion =
$model->empleado->informacionLaboral->catDireccion->nombre_direccion;
$sheet->setCellValue('H10', $nombreDireccion);

$nombreDepartamento =
$model->empleado->informacionLaboral->catDepartamento->nombre_departamento
;
$sheet->setCellValue('H11', $nombreDepartamento);

$nombreTipoContrato =
$model->empleado->informacionLaboral->catTipoContrato->nombre_tipo;
$sheet->setCellValue('H12', $nombreTipoContrato);

//EN BASE AL TIPO DE CONTRATO ASIGNA UN COLOR A LA CELDA
switch ($nombreTipoContrato) {
    case 'Confianza':
```

```
$style = $sheet->getStyle('H12');

$style->getFill()->setFillType(\PhpOffice\PhpSpreadsheet\Style\Fill::FILL_SOLID);
$style->getFill()->getStartColor()->setARGB('FFc7efce'); // Background color #c7efce

$style->getFont()->getColor()->setARGB('FF217346'); // Font color #217346
break;
case 'Sindicalizado':
    $style = $sheet->getStyle('H12');

$style->getFill()->setFillType(\PhpOffice\PhpSpreadsheet\Style\Fill::FILL_SOLID);
$style->getFill()->getStartColor()->setARGB('FFfeeb9d'); // Background color #c7efce

$style->getFont()->getColor()->setARGB('FFa7720f'); // Font color #217346
break;

}

$fecha_permiso = strftime('%A, %B %d, %Y',
strtotime($model->motivoFechaPermiso->fecha_permiso));
$sheet->setCellValue('H14', $fecha_permiso);

$horaSalida = date("g:i A", strtotime($model->hora_salida));
$horaRegreso = date("g:i A", strtotime($model->hora_regreso));
$horaAReponer = date("g:i A", strtotime($model->horario_fecha_a_reponer));

$sheet->setCellValue('H15', $horaSalida);
$sheet->setCellValue('H16', $horaRegreso);

//OBTIENE EL TEXTO INGRESADO EN MOTIVO Y LO LIMPIA
$motivoTextoPlano = strip_tags($model->motivoFechaPermiso->motivo);
$motivoTextoPlano = html_entity_decode($motivoTextoPlano, ENT_QUOTES | ENT_HTML5, 'UTF-8');
$sheet->setCellValue('H18', $motivoTextoPlano);

$fecha_a_reponer = strftime('%A, %B %d, %Y',
strtotime($model->fecha_a_reponer));
$sheet->setCellValue('H20', $fecha_a_reponer);

$sheet->setCellValue('P20', $horaAReponer);
$sheet->setCellValue('A32', $nombreCompleto);

$sheet->setCellValue('A33', $nombrePuesto);
```

```
$nombre = mb_strtoupper($model->empleado->nombre, 'UTF-8');
$apellido = mb_strtoupper($model->empleado->apellido, 'UTF-8');
$profesion = mb_strtoupper($model->empleado->profesion, 'UTF-8');
$nombreCompleto = $profesion.".$apellido . ' ' . $nombre;
$sheet->setCellValue('H32', $nombreCompleto);

//BUSCA EL EMPLEADO QUE ES DIRECTOR
$JuntaGobierno = JuntaGobierno::find()
->where(['nivel_jerarquico' => 'Director'])
->all();

$directorGeneral = null;

foreach ($JuntaGobierno as $Junta) {
    $empleado = Empleado::findOne($Junta->empleado_id);
    //VALIDAD SI SE TRATA DE EL DIRECTOR GENERAL
    if ($empleado && $empleado->informacionLaboral->catPuesto->nombre_puesto ===
        'DIRECTOR GENERAL') {
        $directorGeneral = $empleado;
        break;
    }
}

//SI ES DIRECTOR GENERAL
if ($directorGeneral) {
    $nombre = mb_strtoupper($directorGeneral->nombre, 'UTF-8');
    $apellido = mb_strtoupper($directorGeneral->apellido, 'UTF-8');
    $profesion = mb_strtoupper($directorGeneral->profesion, 'UTF-8');
    $nombreCompleto = $profesion.".$apellido . ' ' . $nombre;
    $sheet->setCellValue('N32', $nombreCompleto);
} else {

}

$sheet->setCellValue('N33', 'DIRECTOR GENERAL');
// Establecer el área de impresión
$htmlWriter = new Html($spreadsheet);
$htmlWriter->setSheetIndex(0);
$htmlWriter->setPreCalculateFormulas(false);
//GENERA EL HTML EN BASE A LA PLANTILLA
$fullHtmlContent = $htmlWriter->generateHtmlAll();

$fullHtmlContent = str_replace(' ', ' ', $fullHtmlContent);
//RENDERIZA LA VISTA PREVIA DEL HTML GENERADO Y MANDA EL
CONTENIDO A ESTA VISTA
```

```
return $this->render('excel-html', ['htmlContent' => $fullHtmlContent, 'model' =>
$model]);
}

public function actionExportHtml($id)
{
    $this->layout = false;

    $model = PermisoFueraTrabajo::findOne($id);

    if (!$model) {
        throw new NotFoundHttpException('El registro no existe.');
    }

    $templatePath = Yii::getAlias('@app/templates/permiso_fuera_trabajo.xlsx');

    $spreadsheet = IOFactory::load($templatePath);

    $sheet = $spreadsheet->getActiveSheet();
    $sheet->setCellValue('F6', $model->empleado->numero_empleado);

    $nombre = mb_strtoupper($model->empleado->nombre, 'UTF-8');
    $apellido = mb_strtoupper($model->empleado->apellido, 'UTF-8');
    $nombreCompleto = $apellido . ' ' . $nombre;
    $sheet->setCellValue('H7', $nombreCompleto);

    setlocale(LC_TIME, 'es_419.UTF-8');
    $fechaHOY = strftime('%A, %B %d, %Y');
    $sheet->setCellValue('N6', $fechaHOY);

    $nombrePuesto =
$model->empleado->informacionLaboral->catPuesto->nombre_puesto;
$sheet->setCellValue('H8', $nombrePuesto);

$nombreCargo =
$model->empleado->informacionLaboral->catDptoCargo->nombre_dpto;
$sheet->setCellValue('H9', $nombreCargo);

$nombreDireccion =
$model->empleado->informacionLaboral->catDireccion->nombre_direccion;
```

```
$sheet->setCellValue('H10', $nombreDireccion);

$nombreDepartamento =
$model->empleado->informacionLaboral->catDepartamento->nombre_departamento
;
$sheet->setCellValue('H11', $nombreDepartamento);

$nombreTipoContrato =
$model->empleado->informacionLaboral->catTipoContrato->nombre_tipo;
$sheet->setCellValue('H12', $nombreTipoContrato);

switch ($nombreTipoContrato) {
    case 'Confianza':
        $style = $sheet->getStyle('H12');

        $style->getFill()->setFillType(\PhpOffice\PhpSpreadsheet\Style\Fill::FILL_SOLID);
        $style->getFill()->getStartColor()->setARGB('FFc7efce'); // Background color #c7efce

        $style->getFont()->getColor()->setARGB('FF217346'); // Font color #217346
            break;
    case 'Sindicalizado':
        $style = $sheet->getStyle('H12');

        $style->getFill()->setFillType(\PhpOffice\PhpSpreadsheet\Style\Fill::FILL_SOLID);
        $style->getFill()->getStartColor()->setARGB('FFfeeb9d'); // Background color #c7efce

        $style->getFont()->getColor()->setARGB('FFa7720f'); // Font color #217346
            break;
}

$fecha_permiso = strftime('%A, %B %d, %Y',
strtotime($model->motivoFechaPermiso->fecha_permiso));
$sheet->setCellValue('H14', $fecha_permiso);
$horaSalida = date("g:i A", strtotime($model->hora_salida));
$horaRegreso = date("g:i A", strtotime($model->hora_regreso));
$horaAReponer = date("g:i A", strtotime($model->horario_fecha_a_reponer));
$sheet->setCellValue('H15', $horaSalida);
$sheet->setCellValue('H16', $horaRegreso);
$motivoTextoPlano = strip_tags($model->motivoFechaPermiso->motivo);
$motivoTextoPlano = html_entity_decode($motivoTextoPlano, ENT_QUOTES | ENT_HTML5, 'UTF-8');
$sheet->setCellValue('H18', $motivoTextoPlano);
```

```
$fecha_a_reponer = strftime('%A, %B %d, %Y',
strtotime($model->fecha_a_reponer));
$sheet->setCellValue('H20', $fecha_a_reponer);

$sheet->setCellValue('P20', $horaAReponer);
$sheet->setCellValue('A32', $nombreCompleto);

$sheet->setCellValue('A33', $nombrePuesto);

$direccion =
CatDireccion::findOne($model->empleado->informacionLaboral->cat_direccion_id);

if ($direccion && $direccion->nombre_direccion !== '1.- GENERAL' ) {
    $nombreJefe =
mb_strtoupper($model->empleado->informacionLaboral->juntaGobierno->empleado-
>nombre, 'UTF-8');
    $apellidoJefe =
mb_strtoupper($model->empleado->informacionLaboral->juntaGobierno->empleado-
>apellido, 'UTF-8');
    $profesionJefe =
mb_strtoupper($model->empleado->informacionLaboral->juntaGobierno->empleado-
>profesion, 'UTF-8');
    $nombreCompletoJefe = $profesionJefe . ' ' . $apellidoJefe . ' ' . $nombreJefe;

    $sheet->setCellValue('H32', $nombreCompletoJefe);
} else {
    $sheet->setCellValue('H32', null);
}

$ juntaDirectorDireccion = JuntaGobierno::find()
->where(['cat_direccion_id' =>
$model->empleado->informacionLaboral->cat_direccion_id])
->andWhere(['or', ['nivel_jerarquico' => 'Director'], ['nivel_jerarquico' => 'Jefe de
unidad']])
->one();

if ($juntaDirectorDireccion) {
```

```
$nombreDirector = mb_strtoupper($juntaDirectorDireccion->empleado->nombre,
'UTF-8');
$apellidoDirector = mb_strtoupper($juntaDirectorDireccion->empleado->apellido,
'UTF-8');
$profesionDirector = mb_strtoupper($juntaDirectorDireccion->empleado->profesion,
'UTF-8');
$nombreCompletoDirector = $profesionDirector . ' ' . $apellidoDirector . ' ' .
$nombreDirector;

$sheet->setCellValue('N32', $nombreCompletoDirector);
//VALIDACIONES PARA IDENTIFICAR LA ESTRUCTURA EN LA QUE SE
LLENARA LA PLANTILLA EN BASE A LA DIRECCION PERTENECIENTE DEL
EMPLEADO
$nombreDireccion = $juntaDirectorDireccion->catDireccion->nombre_direccion;
switch ($nombreDireccion) {
    case '1.- GENERAL':
        if ($model->empleado->informacionLaboral->juntaGobierno->nivel_jerarquico
== 'Jefe de unidad') {
            $nombreJefe =
mb_strtoupper($model->empleado->informacionLaboral->juntaGobierno->empleado-
>nombre, 'UTF-8');
            $apellidoJefe =
mb_strtoupper($model->empleado->informacionLaboral->juntaGobierno->empleado-
>apellido, 'UTF-8');
            $profesionJefe =
mb_strtoupper($model->empleado->informacionLaboral->juntaGobierno->empleado-
>profesion, 'UTF-8');
            $nombreCompletoJefe = $profesionJefe . ' ' . $apellidoJefe . ' ' .
$nombreJefe;

$sheet->setCellValue('N32', $nombreCompletoJefe);
        $tituloDireccion = 'JEFE DE ' .
$model->empleado->informacionLaboral->juntaGobierno->empleado->informacionLa-
boral->catDepartamento->nombre_departamento;
    } else {
        $tituloDireccion = 'DIRECTOR GENERAL prueba';
    }
    break;
    case '2.- ADMINISTRACIÓN':
        $tituloDireccion = 'DIRECTOR DE ADMINISTRACIÓN';
    break;
    case '4.- OPERACIONES':
        $tituloDireccion = 'DIRECTOR DE OPERACIONES';
    break;
    case '3.- COMERCIAL':
        $tituloDireccion = 'DIRECTOR COMERCIAL';
```

```
break;
case '5.- PLANEACION':
    $tituloDireccion = 'DIRECTOR DE PLANEACION';
    break;
default:
    $tituloDireccion = "";
}

$sheet->setCellValue('N33', $tituloDireccion);
} else {
$sheet->setCellValue('N32', null);
$sheet->setCellValue('N33', null);
}
// Establecer el área de impresión
$htmlWriter = new Html($spreadsheet);
$htmlWriter->setSheetIndex(0);
$htmlWriter->setPreCalculateFormulas(false);

$fullHtmlContent = $htmlWriter->generateHtmlAll();

$fullHtmlContent = str_replace('&nbsp;', ' ', $fullHtmlContent);

return $this->render('excel-html', ['htmlContent' => $fullHtmlContent, 'model' => $model]);
}
```

### - Avisos.

Descripción. Controlador que se encarga de crear y mostrar avisos a los usuarios del sistema. En este controlador tenemos la función que permite obtener la imagen asociada al registro del aviso creado, esta función permite acceder al directorio de runtime y dentro de la carpeta imagenes\_avisos, extraer esa imagen y poderla mostrar. También se tiene la función para crear el registro del aviso y manejar la subida de la imagen al directorio correspondiente. La función de update permite editar un registro ya creado y la función de delete borrar tal registro.

```
//FUNCIÓN QUE PERMITE EXTRAER Y MOSTRAR LA IMAGEN DEL
REGISTRO
public function actionVerImagen($nombre)
{
    //OBTIENE LA RUTA Y BUSCA LA IMAGEN
```

```
$filePath = Yii::getAlias('@runtime/imagenes_avisos/' . $nombre);

if (file_exists($filePath)) {
    return Yii::$app->response->sendFile($filePath);
} else {
    throw new \yii\web\NotFoundHttpException('La imagen no existe.');
}

//SE ENCARGA DE MOSTRAR TODO LOS REGISTROS DE AVISOS
public function actionCarruselAvisos()
{
    $avisos = \app\models\Aviso::find()->all();

    return $this->render('carrusel-avisos', [
        'avisos' => $avisos,
    ]);
}

//FUNCION QUE CREA EL REGISTRO DEL AVISO Y SE ENCARGA DE
LA SUBIDA DE IMAGENES
public function actionCreate()
{
    $model = new Aviso();

    if ($model->load(Yii::$app->request->post())) {
        // Manejar la subida de la imagen
        $image = UploadedFile::getInstance($model, 'imagen');

        if ($image) {
            // Crear la carpeta si no existe
            $directory = Yii::getAlias('@runtime/imagenes_avisos');
            if (!is_dir($directory)) {
                FileHelper::createDirectory($directory);
            }

            // Generar un nombre único para la imagen
            $fileName = uniqid() . '.' . $image->extension;
            $filePath = $directory . '/' . $fileName;

            // Guardar la imagen en la carpeta
        }
    }
}
```

```
if ($image->saveAs($filePath)) {
    // Asigna el nombre del archivo al modelo
    $model->imagen = $fileName;
} else {
    // Manejar el error si la imagen no se puede guardar
    $model->addError('imagen', 'No se pudo subir la imagen.');

}

// Guardar el modelo si no hay errores
if ($model->save()) {
    Yii::$app->session->setFlash('success', 'Aviso Publicado.');

    return $this->redirect(['//site/portalgestionrh']);
}
else{

    Yii::$app->session->setFlash('error', 'No se pudo crear el aviso.');

}

return $this->render('create', [
    'model' => $model,
]);
}

public function actionUpdate($id)
{
    $model = $this->findModel($id);

    // Guardar la imagen actual para referencia si no se sube una nueva
    $currentImage = $model->imagen;

    // Manejar la carga del formulario
    if ($model->load(Yii::$app->request->post())) {
        // Manejar la subida de la nueva imagen si existe
        $image = UploadedFile::getInstance($model, 'imagen');

        if ($image) {
            // Crear la carpeta si no existe
```

```
$directory = Yii::getAlias('@runtime/imagenes_avisos');
if (!is_dir($directory)) {
    FileHelper::createDirectory($directory);
}

// Generar un nombre único para la imagen
$fileName = uniqid() . '.' . $image->extension;
$filePath = $directory . '/' . $fileName;

// Guardar la nueva imagen en la carpeta
if ($image->saveAs($filePath)) {
    // Eliminar la imagen anterior si existe
    if ($currentImage && file_exists($directory . '/' . $currentImage)) {
        unlink($directory . '/' . $currentImage);
    }
    // Asigna el nombre del archivo al modelo
    $model->imagen = $fileName;
} else {
    // Manejar el error si la imagen no se puede guardar
    $model->addError('imagen', 'No se pudo subir la imagen.');
    // Si hubo un error, se mantendrá la imagen actual
    $model->imagen = $currentImage;
}
} else {
    // Si no se subió una nueva imagen, mantener la imagen actual
    $model->imagen = $currentImage;
}

// Guardar el modelo si no hay errores
if ($model->save()) {
    Yii::$app->session->setFlash('success', 'Aviso Actualizado');

    return $this->redirect(['//site/portalgestionrh']);
}
else{
    Yii::$app->session->setFlash('error', 'No se puedo actualizar el
aviso .');

}
```

```
        return $this->render('update', [
            'model' => $model,
        ]);
    }

//FUNCIÓN PARA ELIMINAR EL REGISTRO ASOCIADO
public function actionDelete($id)
{
    $model = $this->findModel($id);

    if ($model->delete()) {
        // Puedes agregar una notificación de éxito aquí si lo deseas
        Yii::$app->session->setFlash('success', 'Se quito el aviso
correctamente.');

    } else {
        // Manejar el error si el modelo no se puede eliminar
        Yii::$app->session->setFlash('error', 'No se puedo eliminar el aviso .');

    }
}

return $this->redirect(['//site/portalgestionrh']);
```

## Extensiones.

El sistema ha sido desarrollado utilizando el framework Yii2 en conjunto con Composer, lo que permitió la integración de múltiples extensiones para asegurar el correcto funcionamiento del sistema. Además, una de estas extensiones fue utilizada como plantilla base para desarrollar el diseño del sistema.

## COMPOSER.

En el archivo composer.json, se incluyeron las siguientes dependencias necesarias para el funcionamiento del sistema:

```
"php": ">=7.4.0",
"yiisoft/yii2": "~2.0.45",
"yiisoft/yii2-bootstrap5": "~2.0.2",
"yiisoft/yii2-symfonymailer": "~2.0.3",
"hail812/yii2-adminlte3": "~1.1",
```

```
"almasaeed2010/adminlte": "~3.1",
"hail812/yii2-adminlte-widgets": "~1.0.2",
"ktaris/yii2-mexvalidators": "0.0.1",
"kartik-v/yii2-widget-datepicker": "dev-master",
"yiisoft/yii2-jui": "^2.0",
"yiisoft/yii2-swiftmailer": "^2.0",
"kartik-v/yii2-widget-fileinput": "dev-master",
"kartik-v/yii2-field-range": "dev-master",
"kartik-v/yii2-detail-view": "dev-master",
"kartik-v/yii2-widget-select2": "dev-master",
"kartik-v/yii2-bootstrap5-dropdown": "dev-master",
"kartik-v/yii2-tabs-x": "dev-master",
"kartik-v/yii2-date-range": "dev-master",
"kartik-v/yii2-widget-timepicker": "dev-master",
"kartik-v/yii2-widget-activeform": "**",
"dompdf/dompdf": "^3.0",
"phpoffice/phpspreadsheet": "^2.1",
"tecnickcom/tcpdf": "^6.7",
"mpdf/mpdf": "^8.2",
"edofre/yii2-fullcalendar": "^1.0",
"bizley/quill": "^3.3",
"phpoffice/phpword": "^1.2",
"box/spout": "^3.3",
"froala/yii2-froala-editor": "^4.2",
"mdmsoft/yii2-admin": "~2.0",
"vumanskyi/yii2-seo-breadcrumbs": "^1.0",
"jlorente/yii2-widget-remainingcharacters": "**"
```

## RBAC.

RBAC, por sus siglas en inglés (Role-Based Access Control), es uno de los métodos de autorización que ofrece Yii2, junto con ACF (Access Control Filter). En este sistema se implementó RBAC utilizando la extensión mdmsoft/yii2-admin (~2.0), lo que facilita un control de usuarios más eficiente. Esta extensión permite la creación de roles, asignación de permisos, gestión de rutas permitidas, entre otras funcionalidades, proporcionando un entorno seguro y personalizado para el acceso de usuarios.

Más información sobre su instalación:

<https://tutoriales-yii2.blogspot.com/2020/03/como-instalar-rbac.html>

### AdminLTE3.

Para el diseño de la interfaz de usuario, se utilizó la plantilla AdminLTE3, implementada mediante la extensión hail812/yii2-adminlte3 (~1.1). Esta plantilla proporciona un diseño moderno y responsive, que facilita la creación de paneles de administración con una estructura limpia y profesional.

La extensión permite integrar fácilmente los componentes de AdminLTE con Yii2, proporcionando widgets, estilos, y scripts que mejoran la experiencia del usuario y permiten una personalización sencilla del diseño. Además, se utilizaron otras extensiones relacionadas como hail812/yii2-adminlte-widgets y almasaeed2010/adminlte para complementar la funcionalidad y aprovechar al máximo las características de AdminLTE3.

Más información sobre su instalación:

<https://www.yiiframework.com/extension/hail812/yii2-adminlte3>

### Estilos.

El diseño de este sistema se basa principalmente en los estilos proporcionados por la plantilla AdminLTE3, así como en los estilos de Bootstrap, tanto en su versión 4 como en la versión 5. AdminLTE3 también ofrece una variedad de widgets que se han implementado en diferentes partes del sistema para mejorar la interfaz y la experiencia del usuario.

Además de los estilos integrados por estas bibliotecas, se han utilizado hojas de estilo personalizadas. Estas hojas de estilo se encuentran en el directorio web/css y se han utilizado para ajustar aspectos específicos del diseño que no estaban cubiertos por los estilos predeterminados de AdminLTE3 o Bootstrap.

## Conclusión.

El objetivo de este manual es proporcionar una guía detallada para que los desarrolladores comprendan la estructura y el funcionamiento completo del sistema. Es fundamental consultar este manual antes de realizar cualquier cambio, ya que contiene información clave sobre la definición de permisos y otros aspectos cruciales que deben ser considerados para futuras actualizaciones, mejoras, y para mantener el sistema en orden y funcionando correctamente. También se detallan los requisitos necesarios para su correcta instalación.

Además, el manual ofrece un análisis detallado de la estructura de la base de datos, incluyendo las tablas, atributos, tipos de datos, y las relaciones entre ellas, lo que facilita la comprensión del flujo de datos dentro del sistema.

Aunque el sistema contiene una cantidad considerable de tablas, vistas, modelos y controladores con múltiples funciones, cada uno de estos componentes es fácil de entender debido a su lógica similar. No obstante, es importante recordar que el sistema está desarrollado en el framework Yii2, por lo que, en caso de requerir un entendimiento más profundo, se recomienda consultar la documentación oficial del framework y sus herramientas.