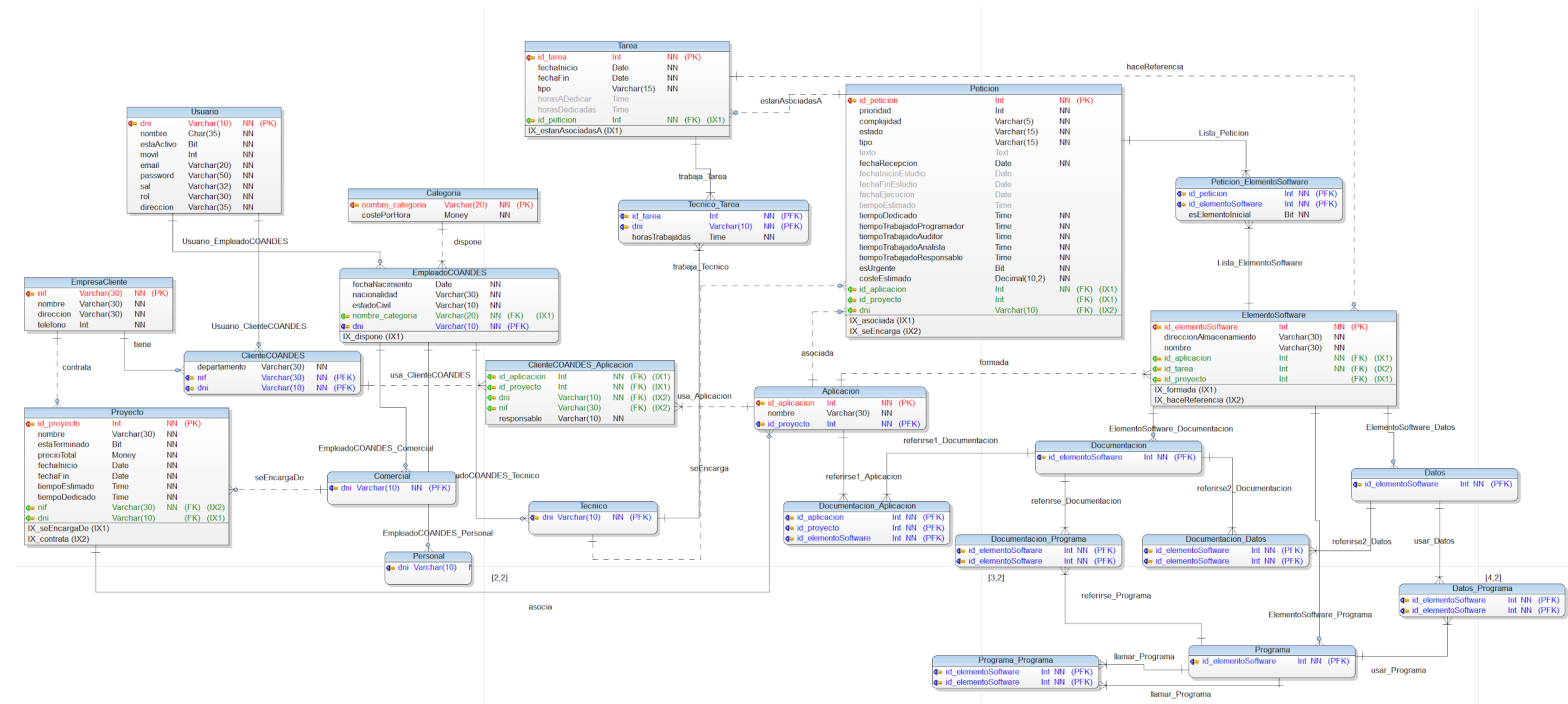


Contenido

6. Modelo de datos	1
6.1. Diagramas de tablas	1
6.1.1. Explicación del diagrama	1
6.1.2. Script BD	5
6.2. Otros elementos de almacenamiento	15
6.2.1. Diseño y explicación del fichero XML	15
6.2.2. Diseño y explicación de la plantilla de recuperar contraseña.	17
6.2.3. Diseño y explicación del SMS de confirmación.	17
6.2.4. Diseño y explicación del correo de confirmación de registro.	17
6.2.3. Diseño y explicación del correo con una contraseña generada.	18





6. Modelo de datos

6.1. Diagramas de tablas







6.1.1. Explicación del diagrama

Desnormalización:

Peticion		
 id_peticion	Int	NN (PK)
prioridad	Int	NN
complejidad	Varchar(5)	NN
estado	Varchar(15)	NN
tipo	Varchar(15)	NN
texto	Text	
fechaRecepcion	Date	NN
fechaInicioEstudio	Date	
fechaFinEstudio	Date	
fechaEjecucion	Date	
tiempoEstimado	Time	
tiempoDedicado	Time	NN
tiempoTrabajadoProgramador	Time	NN
tiempoTrabajadoAuditor	Time	NN
tiempoTrabajadoAnalista	Time	NN
tiempoTrabajadoResponsable	Time	NN
esUrgente	Bit	NN
 id_aplicacion	Int	NN (FK) (IX1)
 id_proyecto	Int	(FK) (IX1)
 dni	Int	(FK) (IX2)
IX_asociada (IX1)		
IX_seEncarga (IX2)		

Debido al coste elevado en la obtención de los tiempos totales de cada trabajador según la categoría se ha decidido desnormalizar en la tabla petición añadiendo los campos: tiempoTrabajadorProgramador, tiempoTrabajadorAuditor, tiempoTrabajadoAnalista y tiempoTrabajadoResponsable. Estos campos se guardan la suma del tiempo empleado por los empleados de cada categoría que están trabajando en tareas de esa petición.

Peticion		
 id_peticion	Int	NN (PK)
prioridad	Int	NN
complejidad	Varchar(5)	NN
estado	Varchar(15)	NN
tipo	Varchar(15)	NN
texto	Text	
fechaRecepcion	Date	NN
fechaInicioEstudio	Date	
fechaFinEstudio	Date	
fechaEjecucion	Date	
tiempoEstimado	Time	
tiempoDedicado	Time	NN
tiempoTrabajadoProgramador	Time	NN
tiempoTrabajadoAuditor	Time	NN
tiempoTrabajadoAnalista	Time	NN
tiempoTrabajadoResponsable	Time	NN
esUrgente	Bit	NN
costeEstimado	Decimal(10,2)	NN
 id_aplicacion	Int	NN (FK) (IX1)
 id_proyecto	Int	(FK) (IX1)
 dni	Int	(FK) (IX2)
IX_asociada (IX1)		
IX_seEncarga (IX2)		

En petición:

Hemos añadido el campo `costeEstimado`, debido a su alta complejidad, ya que es un dato que requiere de muchos detalles para su creación (tiempo estimado, coste de cada hora, tiempo estimado por cada empleado con sus respectivos costes de hora, porcentaje añadido en caso de ser urgente...) y, el simple hecho de que lo vayamos a utilizar de forma frecuente por ejemplo con la automatización de los cobros al cliente mediante un trigger... hemos decidido añadir el campo.

Relaciones de herencia:

En relaciones de herencia, hemos decidido usar las N tablas en todas las herencias.

Uno de los motivos principales era para garantizar la unicidad de los IDs entre los hijos de la herencia, evitando que puedan existir usuarios con el mismo ID, en diferentes departamentos, también, evitando que existan usuarios clientes y empleados al mismo tiempo.

Crearlos todo en una única tabla no tiene sentido debido a que las relaciones perderían toda su lógica en ese momento, podríamos tener tanto clientes como empleados en lugares en los que no deberían estar, siendo un fallo de seguridad.

Las relaciones 1:1 no ha hecho falta cambiar su orden, debido a que ya las tuvimos en cuenta en el modelo lógico, para que la cardinalidad tenga un sentido más detallado.

Relaciones identificadas:

tiene (EmpresaClientes - ClienteCOANDES): porque tenemos que saber el cliente a qué empresa pertenece, además de que, el cliente solicitará aplicaciones para la empresa en cuestión, siendo la relación identificada se obtendría una clave foránea en la tabla entre ClienteCOANDES y Aplicación, sabiendo en todo momento a qué empresa pertenece cada aplicación.

asocia (Proyecto - Aplicación): En el caso que tuviésemos múltiples aplicaciones iguales en diferentes proyectos, sería mucho más complejo realizar una petición sobre dicha aplicación sin tener el proyecto como clave primaria.

6.1.2. Script BD

/*

Created: 18/12/2020

Modified: 20/12/2020

Project: ProyectoTOAD_ISA2020

Model: Physical model

Database: MySQL 8.0

*/

-- Create tables section -----

-- Table ClienteCOANDES

CREATE TABLE `ClienteCOANDES`

(

 `departamento` Varchar(30) NOT NULL,

 `dni_usuario` Int NOT NULL

);

ALTER TABLE `ClienteCOANDES` ADD PRIMARY KEY (`dni_usuario`);

-- Table Proyecto

CREATE TABLE `Proyecto`

(

 `id_proyecto` Int NOT NULL,

 `nombre` Varchar(30) NOT NULL,

 `estaTerminado` Bool NOT NULL,

 `precioTotal` Decimal(10,2) NOT NULL,

 `fechaInicio` Date NOT NULL,

 `fechaFin` Date NOT NULL,

```
`tiempoEstimado` Time NOT NULL,  
`tiempoDedicado` Time NOT NULL,  
`dni_usuario` Int  
);
```

```
CREATE INDEX `IX_seEncargaDe` ON `Proyecto` (`dni_usuario`);
```

```
ALTER TABLE `Proyecto` ADD PRIMARY KEY (`id_proyecto`);
```

```
-- Table Aplicacion
```

```
CREATE TABLE `Aplicacion`  
(  
  `id_aplicacion` Int NOT NULL,  
  `nombre` Varchar(30) NOT NULL,  
  `id_proyecto` Int NOT NULL,  
  `dni_usuario` Int  
);
```

```
CREATE INDEX `IX_usa` ON `Aplicacion` (`dni_usuario`);
```

```
ALTER TABLE `Aplicacion` ADD PRIMARY KEY (`id_aplicacion`, `id_proyecto`);
```

```
-- Table ElementoSoftware
```

```
CREATE TABLE `ElementoSoftware`  
(  
  `id_elementoSoftware` Int NOT NULL,  
  `direccionAlmacenamiento` Varchar(30) NOT NULL,  
  `nombre` Varchar(30) NOT NULL,  
  `id_tecnicoCOANDES` Int NOT NULL,
```

```
`id_aplicacion` Int NOT NULL,  
`id_tarea` Int NOT NULL,  
`id_proyecto` Int  
);
```

```
CREATE INDEX `IX_formada` ON `ElementoSoftware` (`id_aplicacion`, `id_proyecto`);
```

```
CREATE INDEX `IX_haceReferencia` ON `ElementoSoftware` (`id_tarea`);
```

```
ALTER TABLE `ElementoSoftware` ADD PRIMARY KEY (`id_elementoSoftware`);
```

-- Table Datos

```
CREATE TABLE `Datos`  
(  
  `id_elementoSoftware` Int NOT NULL  
);
```

```
ALTER TABLE `Datos` ADD PRIMARY KEY (`id_elementoSoftware`);
```

-- Table Programa

```
CREATE TABLE `Programa`  
(  
  `id_elementoSoftware` Int NOT NULL  
);
```

```
ALTER TABLE `Programa` ADD PRIMARY KEY (`id_elementoSoftware`);
```

-- Table Documentacion

```
CREATE TABLE `Documentacion`  
(  
  `id_elementoSoftware` Int NOT NULL  
);
```

```
ALTER TABLE `Documentacion` ADD PRIMARY KEY (`id_elementoSoftware`);
```

-- Table Usuario

```
CREATE TABLE `Usuario`  
(  
  `dni_usuario` Int NOT NULL,  
  `nombre` Char(35) NOT NULL,  
  `estaActivo` Bool NOT NULL,  
  `movil` Int NOT NULL,  
  `email` Varchar(20) NOT NULL,  
  `password` Varchar(50) NOT NULL,  
  `sal` Varchar(32) NOT NULL,  
  `rol` Varchar(30) NOT NULL,  
  `direccion` Varchar(35) NOT NULL  
);
```

```
ALTER TABLE `Usuario` ADD PRIMARY KEY (`dni_usuario`);
```

-- Table Peticion

```
CREATE TABLE `Peticion`  
(  
  `id_peticion` Int NOT NULL,  
  `prioridad` Int NOT NULL,  
  `complejidad` Varchar(5) NOT NULL,
```

```

`estado` Varchar(15) NOT NULL,
`tipo` Varchar(15) NOT NULL,
`texto` Longtext,
`fechaRecepcion` Date NOT NULL,
`fechaInicioEstudio` Date,
`fechaFinEstudio` Date,
`fechaEjecucion` Date,
`tiempoEstimado` Time,
`tiempoDedicado` Time NOT NULL,
`esUrgente` Bool NOT NULL,
`tiempoTrabajadoProgramador` Time NOT NULL,
`tiempoTrabajadoAuditor` Time NOT NULL,
`tiempoTrabajadoAnalista` Time NOT NULL,
`tiempoTrabajadoResponsable` Time NOT NULL,
`costeEstimado` Decimal(10,2) NOT NULL,
`id_aplicacion` Int NOT NULL,
`id_proyecto` Int,
`dni_usuario_tecnico` Int,
`dni_usuario` Int
);

```

```

CREATE INDEX `IX_asociada` ON `Peticion` (`id_aplicacion`, `id_proyecto`);

```

```

CREATE INDEX `IX_seEncarga1` ON `Peticion` (`dni_usuario_tecnico`);

```

```

CREATE INDEX `IX_seEncarga` ON `Peticion` (`dni_usuario`);

```

```

ALTER TABLE `Peticion` ADD PRIMARY KEY (`id_peticion`);

```

```

-- Table Tarea

```



```
CREATE TABLE `Tarea`  
(  
  `id_tarea` Int NOT NULL,  
  `fechaInicio` Date NOT NULL,  
  `fechaFin` Date NOT NULL,  
  `tipo` Varchar(15) NOT NULL,  
  `horasADedicar` Time,  
  `horasDedicadas` Time,  
  `id_peticion` Int NOT NULL,  
  `dni_usuario` Int  
);
```

```
CREATE INDEX `IX_estanAsociadasA` ON `Tarea` (`id_peticion`);
```

```
CREATE INDEX `IX_trabaja` ON `Tarea` (`dni_usuario`);
```

```
ALTER TABLE `Tarea` ADD PRIMARY KEY (`id_tarea`);
```

```
-- Table EmpleadoCOANDES
```

```
CREATE TABLE `EmpleadoCOANDES`  
(  
  `fechaNacimiento` Date NOT NULL,  
  `nacionalidad` Varchar(30) NOT NULL,  
  `estadoCivil` Varchar(10) NOT NULL,  
  `nombre_categoria` Varchar(20) NOT NULL,  
  `dni_usuario` Int NOT NULL  
);
```

```
CREATE INDEX `IX_dispone` ON `EmpleadoCOANDES` (`nombre_categoria`);
```

```
ALTER TABLE `EmpleadoCOANDES` ADD PRIMARY KEY (`dni_usuario`);
```

```
-- Table Categoria
```

```
CREATE TABLE `Categoria`  
(  
  `nombre_categoria` Varchar(20) NOT NULL,  
  `costePorHora` Decimal(10,2) NOT NULL  
);
```

```
ALTER TABLE `Categoria` ADD PRIMARY KEY (`nombre_categoria`);
```

```
-- Table Comercial
```

```
CREATE TABLE `Comercial`  
(  
  `dni_usuario` Int NOT NULL  
);
```

```
ALTER TABLE `Comercial` ADD PRIMARY KEY (`dni_usuario`);
```

```
-- Table Tecnico
```

```
CREATE TABLE `Tecnico`  
(  
  `dni_usuario` Int NOT NULL  
);
```

```
ALTER TABLE `Tecnico` ADD PRIMARY KEY (`dni_usuario`);
```

```
-- Table Personal
```

```
CREATE TABLE `Personal`
```

```
(  
  `dni_usuario` Int NOT NULL  
);
```

```
ALTER TABLE `Personal` ADD PRIMARY KEY (`dni_usuario`);
```

```
-- Table Programa_Programa
```

```
CREATE TABLE `Programa_Programa`
```

```
(  
  `id_elementoSoftware` Int NOT NULL,  
  `id_elementoSoftware_programa` Int NOT NULL  
);
```

```
-- Table Datos_Programa
```

```
CREATE TABLE `Datos_Programa`
```

```
(  
  `id_elementoSoftware` Int NOT NULL,  
  `id_elementoSoftware_programa` Int NOT NULL  
);
```

```
-- Table Documentacion_Aplicacion
```

```
CREATE TABLE `Documentacion_Aplicacion`
```

```
(  
  `id_aplicacion` Int NOT NULL,  
  `id_proyecto` Int NOT NULL,  
  `id_elementoSoftware` Int NOT NULL
```

```
);
```

```
-- Table Documentacion_Programa
```

```
CREATE TABLE `Documentacion_Programa`
```

```
(
```

```
  `id_elementoSoftware` Int NOT NULL,
```

```
  `id_elementoSoftware_documentacion` Int NOT NULL
```

```
);
```

```
-- Table Documentacion_Datos
```

```
CREATE TABLE `Documentacion_Datos`
```

```
(
```

```
  `id_elementoSoftware_datos` Int NOT NULL,
```

```
  `id_elementoSoftware` Int NOT NULL
```

```
);
```

```
-- Table Peticion_ElementoSoftware
```

```
CREATE TABLE `Peticion_ElementoSoftware`
```

```
(
```

```
  `id_peticion` Int NOT NULL,
```

```
  `id_elementoSoftware` Int NOT NULL
```

```
);
```

```
-- Create foreign keys (relationships) section -----
```

```
ALTER TABLE `Peticion` ADD CONSTRAINT `asociada` FOREIGN KEY (`id_aplicacion`, `id_proyecto`)  
REFERENCES `Aplicacion` (`id_aplicacion`, `id_proyecto`) ON DELETE RESTRICT ON UPDATE RESTRICT;
```

```
ALTER TABLE `ElementoSoftware` ADD CONSTRAINT `formada` FOREIGN KEY (`id_aplicacion`, `id_proyecto`) REFERENCES `Aplicacion` (`id_aplicacion`, `id_proyecto`) ON DELETE RESTRICT ON UPDATE RESTRICT;
```

```
ALTER TABLE `ElementoSoftware` ADD CONSTRAINT `haceReferencia` FOREIGN KEY (`id_tarea`) REFERENCES `Tarea` (`id_tarea`) ON DELETE RESTRICT ON UPDATE RESTRICT;
```

```
ALTER TABLE `EmpleadoCOANDES` ADD CONSTRAINT `dispone` FOREIGN KEY (`nombre_categoria`) REFERENCES `Categoria` (`nombre_categoria`) ON DELETE RESTRICT ON UPDATE RESTRICT;
```

```
ALTER TABLE `Aplicacion` ADD CONSTRAINT `asocia` FOREIGN KEY (`id_proyecto`) REFERENCES `Proyecto` (`id_proyecto`) ON DELETE RESTRICT ON UPDATE RESTRICT;
```

```
ALTER TABLE `Tarea` ADD CONSTRAINT `estanAsociadasA` FOREIGN KEY (`id_peticion`) REFERENCES `Peticion` (`id_peticion`) ON DELETE RESTRICT ON UPDATE RESTRICT;
```

```
ALTER TABLE `Tarea` ADD CONSTRAINT `trabaja` FOREIGN KEY (`dni_usuario`) REFERENCES `Tecnico` (`dni_usuario`) ON DELETE RESTRICT ON UPDATE RESTRICT;
```

```
ALTER TABLE `Proyecto` ADD CONSTRAINT `seEncargaDe` FOREIGN KEY (`dni_usuario`) REFERENCES `Comercial` (`dni_usuario`) ON DELETE RESTRICT ON UPDATE RESTRICT;
```

```
ALTER TABLE `Peticion` ADD CONSTRAINT `seEncarga1` FOREIGN KEY (`dni_usuario_tecnico`) REFERENCES `Tecnico` (`dni_usuario`) ON DELETE RESTRICT ON UPDATE RESTRICT;
```

```
ALTER TABLE `Peticion` ADD CONSTRAINT `seEncarga` FOREIGN KEY (`dni_usuario`) REFERENCES `Comercial` (`dni_usuario`) ON DELETE RESTRICT ON UPDATE RESTRICT;
```

```
ALTER TABLE `Aplicacion` ADD CONSTRAINT `usa` FOREIGN KEY (`dni_usuario`) REFERENCES `ClienteCOANDES` (`dni_usuario`) ON DELETE RESTRICT ON UPDATE RESTRICT;
```

6.2. Otros elementos de almacenamiento

6.2.1. Diseño y explicación del fichero XML

El fichero XML está formado por una cabecera y el cuerpo del documento junto a las etiquetas.

Cabecera: se empieza con una declaración XML que sirve para declarar el documento como un documento XML

```
<?xml version="1.0" encoding="UTF-8"?>
```

Cuerpo: contiene la raíz `elementossoftware` que es único y a continuación los distintos elementos del software.

Cada elemento tiene una etiqueta definida (`<documentacion>`, `<datos>` y `<programa>`) que tiene su cierre. Dentro de cada una de estas etiquetas se definen los atributos de la entidad padre en la herencia que tiene `ElementoSoftware`. Estos atributos son definidos entre las etiquetas del mismo nombre, como por ejemplo `<id_elementoSoftware>`, `<direccionAlmacenamiento>`, `<nombre>` y `<id_tecnicoCOANDES>`.

También dentro de cada una de las etiquetas de cada elemento están las dependencias (si las hay) y se usa la etiqueta `<dependencia>`. Dependiendo del tipo de dependencia se podrá definir `<programa>`, `<datos>`, `<aplicacion>` y `<documentacion>` cuyo interior almacena el id correspondiente para poder crear la relación entre ellos a la hora de crear la tabla necesaria. Habrá tantas etiquetas como dependencias necesite, al igual que se podrá repetir distintas etiquetas de un tipo de elemento.

Como ejemplo se muestra el formato del cuerpo XML. En este caso hay tres elementos del software: una documentación, unos datos y un programa dentro de la raíz `<elementossoftware>`. En cada uno de estos tres elementos del software tiene sus etiquetas para los atributos que son obligatorios y las dependencias que son opcionales. En las dependencias se ha puesto como ejemplo las distintas dependencias que pueda tener según qué tipo de elemento de software, aunque por supuesto, podrá tener más.

```
<elementossoftware>
```

```
<documentacion>
```

```
    <id_elementoSoftware>{id_elementoSoftware}</id_elementoSoftware>
```

```
    <direccionAlmacenamiento>{direccionAlmacenamiento}</direccionAlmacenamiento>
```

```
    <nombre>{nombre}</nombre>
```

```
    <dependencias>
```

```
        <programa>{id_elementoSoftware}</programa>
```

```
        <datos>{id_elementoSoftware_datos}</datos>
```

```

        <aplicacion>{id_aplicacion}</aplicacion>
    </dependencias>
</documentacion>
<datos>
    <id_elementoSoftware >{id_elementoSoftware}<id_elementoSoftware>
    <direccionAlmacenamiento> {direccionAlmacenamiento} </direccionAlmacenamiento>
    <nombre>{nombre}</nombre>
    <dependencias>
        <programa>{id_elementoSoftware}</programa>
        <documentacion>{id_elementoSoftware}</documentacion>
    </dependencias>
</datos>
<programa>
    <id_elementoSoftware >{id_elementoSoftware}<id_elementoSoftware>
    <direccionAlmacenamiento> {direccionAlmacenamiento} </direccionAlmacenamiento>
    <nombre>{nombre}</nombre>
    <dependencias>
        <programa>{id_elementoSoftware}</programa>
        <documentacion>{id_elementoSoftware}</documentacion>
        <datos>{id_elementoSoftware}</datos>
    </dependencias>
</programa>
</elementossoftware>

```

Explicación textual de su uso: el fichero XML servirá para que los clientes puedan cargar automáticamente en el sistema los elementos de cada aplicación y sus dependencias.

6.2.2. Diseño y explicación de la plantilla de recuperar contraseña.

El mensaje tendrá el siguiente formato:

Vendrá enviado por: noreply@COANDES.es

Título: Correo recuperación contraseña.

Cuerpo: Para recuperar la contraseña, acceda al siguiente enlace:

<https://COANDES.es/recuperarcontrasena/{numerosyletrasgenerados}>

Cierre: Mensaje generado de forma automática, no responda a este mensaje.

Explicación textual de su uso:

El mensaje servirá para cambiar la contraseña actual del usuario registrado con el mismo correo electrónico.

numerosyletrasgenerados garantiza la unicidad del enlace, debido a que cada usuario tendrá su propio enlace para restablecer su propia contraseña.

6.2.3. Diseño y explicación del SMS de confirmación.

El SMS tendrá el siguiente formato:

COANDES: Para realizar la operación, utiliza el código: {codigoGenerado}

Explicación textual de su uso:

El código sirve para el factor de doble verificación, para asegurarnos de que el usuario está accediendo de forma segura a la aplicación.

6.2.4. Diseño y explicación del correo de confirmación de registro.

El correo tendrá el siguiente formato:

Vendrá enviado por: noreply@COANDES.es

Título: Confirmación de registro.

Cuerpo: Haga click en el siguiente enlace para confirmar la creación de una cuenta en <https://COANDES.es/confirmarusuario/{numerosyletrasgenerados}>

Cierre: Mensaje generado de forma automática, no responda a este mensaje.

Explicación textual de su uso:

El mensaje viene enviado al usuario después de hablar con un empleado comercial.

El objetivo del mensaje es confirmar que el usuario quiere registrarse en COANDES.

numerosyletrasgenerados garantiza la unicidad del enlace, debido a que cada usuario tendrá su propio enlace para confirmar la creación de su propia cuenta.

6.2.3. Diseño y explicación del correo con una contraseña generada.

El correo tendrá el siguiente formato:

Vendrá enviado por: noreply@COANDES.es

Título: Contraseña generada.

Cuerpo: La contraseña de su cuenta en COANDES es: {contraseñagenerada}

Cierre: Mensaje generado de forma automática, no responda a este mensaje.

Explicación textual de su uso:

El correo contiene la contraseña necesaria para que el usuario loguee en COANDES.