

# Universidad Nacional

**Escuela de Informática  
Cátedra de Ingeniería de Sistemas**



**Manual Técnico  
Sistema de Correspondencia SITUN  
Código SCC-SITUN  
Número del grupo de proyecto 21  
Versión 1**

**Autores del software:**

Alexandra Aguilar Nájera, Cédula: 3 0478 0037

Ignacio Avalos Bonilla, Cédula: 2 0730 0601

David Chacón Abarca, Cédula 1 1624 0217

Hillary Porras Ríos, Cédula: 1 1634 0968

Bryhan Rodríguez Mora, Cédula: 1 1542 0325

Jean Carlo Vargas Zúñiga, Cédula: 4 0222 0474

## **Tabla de Contenidos**

### 1. Introducción

#### 1.1 Audiencia

#### 1.2 Propósito

#### 1.3 Alcance

### 2. Arquitectura del Sistema

### 4. Plataforma Técnica y Seguridad

### 7. Componentes del Sistema

### 14. Base de datos

## **Introducción**

En este documento se explican y detallan los aspectos técnicos más importantes del Sistema de Correspondencia SITUN, esto para facilitar una mayor comprensión de cómo fue desarrollado. Se inicia con una pequeña explicación mediante diagramas sobre cuál es la arquitectura con la que se construyó el sistema, también se mencionan las herramientas de software que se utilizaron como materia prima para su construcción. Seguidamente se explican los principales componentes así como sus detalles en términos de programación según los lenguajes utilizados, además de una explicación sobre su funciones como partes del sistema. Finalmente se explica cómo fue diseñada y creada la base de datos, así como los componentes que la conforman.

## **Audiencia**

Este manual va dirigido a los usuarios del sistema de la empresa patrocinadora y a él o los encargados del mantenimiento del sistema, pero principalmente hacia estos últimos, ya que pueden darse distintas circunstancias a futuro en donde se requiera consultar como se desarrolló el sistema y sus componentes, para ya sea realizar mejoras en el sistema o dar mantenimiento en alguno de sus módulos.

## **Propósito**

El propósito de este documento es dar una visión más amplia de la arquitectura del sistema y cuáles son sus componentes.

## **Alcance**

Esta versión del manual técnico cubre los aspectos técnicos detallados del sistema Control de Correspondencia SITUN, con relación a la arquitectura del software cubriendo los siguientes aspectos:

- **Arquitectura:** En este apartado se inicia con una fase introductoria del manual en donde se explica cuál es la arquitectura que utiliza el sistema y su funcionamiento.
- **Plataforma técnica y Seguridad:** En esta sección se muestra se habla sobre las herramientas de software utilizadas para el desarrollo del sistema.
- **Componentes del Sistema:** En esta sección se muestra cómo está dividido el sistema software en módulos y las dependencias que hay entre esos componentes.

- Base de datos: En esta sección se detallan el modelo de bases de datos utilizados, así como su representación gráfica en un diagrama. También se hace mención a sus tablas con sus respectivos atributos, además de información sobre las funciones almacenadas existentes.

## Arquitectura del Sistema

La Arquitectura que se utiliza en este sistema es del tipo Multicapa la cual se encuentra dividida en 3 capas: Capa de Presentación, Capa de Negocio, Capa de Gestión de datos. Este modelo busca la separación de la lógica con el diseño, con esto se contribuye a la comprensión y organización del sistema, finalmente ofrece una enorme flexibilidad en el diseño de la aplicación. Es importante mencionar que esta arquitectura fue diseñada para trabajar en forma de cliente servidor lo cual se ajusta también a las necesidades del desarrollo del proyecto. La arquitectura a utilizar se divide en 3 capas como se muestra en la siguiente figura:

**Figura 1.** Capas de Arquitectura



**Fuente:** Elaboración Propia

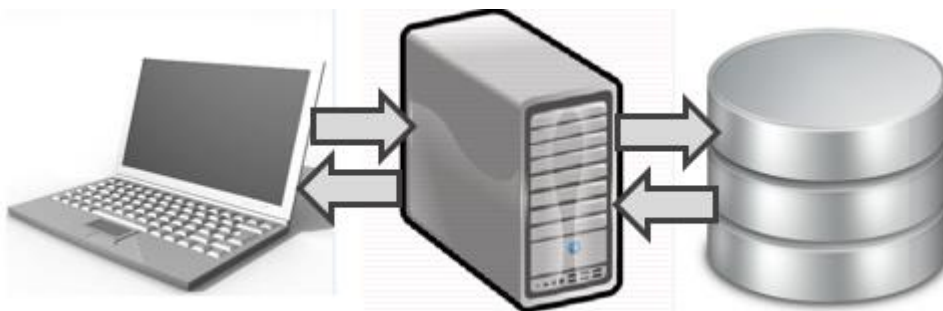
Las capas de la figura anterior están en compuestas por:

Capa de Presentación: Esta capa cuenta con la interfaz para interactuar con el usuario en este caso los empleados del SITUN encargados de la correspondencia que ingresa al SITUN. En la cual se mostrarán todas las opciones del sistema que responden a los requerimientos funcionales y no funcionales del mismo. Los cuales se detallaran más a fondo en este documento en las distintas vistas de la arquitectura.

- Capa de Negocio: Esta capa contiene la parte lógica donde estarán todas las funciones y servicios que ejecuta el Servidor. También en donde se reciben las peticiones y se procesa la información para enviar respuestas a las peticiones del usuario.
- Capa de Gestión de datos: Finalmente esta capa contiene la persistencia de los datos que requiere almacenar el sistema, por consiguiente la base de datos ubicada en este nivel se encarga de gestionar la información.

A continuación se muestra un diagrama de la forma en que trabaja esta arquitectura en este caso cliente-servidor como se mencionó anteriormente al inicio de este apartado.

**Figura 2.** Procesos entre capas de la arquitectura



**Fuente: Elaboración Propia**

En el diagrama anterior se puede observar como el cliente enviará peticiones al servidor y por consiguiente el servidor le enviará de vuelta al cliente las respuestas a sus peticiones. También el servidor trabaja de manera conjunta con la base de

datos para realizar peticiones de almacenamiento y recibir respuesta a peticiones de obtención de datos.

## **Plataforma Técnica y Seguridad**

El sistema planteado es ejecutado sobre una plataforma web. Antes de iniciar con el detalle de las herramientas utilizadas es importante mencionar que el sistema fue desarrollado a base de componentes de software libre, lo cual quiere decir que no depende del pago de licencias por parte de la empresa patrocinadora. Dentro de las herramientas de software utilizadas se encuentran:

- **HTML:** sus siglas en inglés HyperText Markup Language (lenguaje de marcas de hipertexto). hace referencia al lenguaje de etiquetas el cual se utiliza para la elaboración de páginas web. Es un estándar que sirve de referencia del software que conecta con la elaboración de páginas web en sus diferentes versiones, define una estructura básica y un código (denominado código HTML) para la definición de contenido de una página web, como texto, imágenes, videos, juegos, entre otros. Su función dentro del sistema es la de proporcionar las vistas al usuario y sus componentes, es utilizado en conjunto con otros elementos que se explicaran más adelante.
- **Javascript:** Es un lenguaje de programación interpretado. Se define como objetos. Se utiliza principalmente en su forma del lado del cliente (Front-End), implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas. Su función en este sistema es la de manejo de funciones que requieran el envío o petición de datos hacia la parte del Back-End ( Servidor , Base de Datos ) , sin embargo es importante mencionar de nuevo que trabaja desde la parte del navegador en este caso en conjunto con las páginas HTML.
- **Node.js:** Es un entorno de ejecución para JavaScript construido con el motor de JavaScript V8 de Chrome. En este caso se utiliza esta herramienta principalmente para la construcción del servidor, así como el manejo de peticiones y envío de datos desde el servidor.
- **Jquery:** Es una biblioteca multiplataforma de JavaScript, creada inicialmente por John Resig, que permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones. Para este caso el uso de jquery se utiliza para simplificar código, funcionalidades en el diseño HTML, validación de formularios.

- Bootstrap: Es un framework originalmente creado por Twitter, es que permite crear interfaces web con CSS y JavaScript. Para el sistema se aprovechó la característica que posee de adaptar la interfaz del sitio web al tamaño del dispositivo en que se visualice. Es decir, el sitio web se adapta automáticamente al tamaño de una PC, una Tablet u otro dispositivo. Esta técnica de diseño y desarrollo se conoce como “responsive design” o diseño adaptativo. El beneficio de usar responsive design en un sitio web, es principalmente que el sitio web se adapta automáticamente al dispositivo desde donde se acceda.  
También entre sus funciones para el sistema se utiliza para dar un diseño más estética a las vistas que se le ofrecen al usuario en las páginas HTML.
- Angular: AngularJS es un framework MVC de JavaScript para el Desarrollo Web Front End que permite crear aplicaciones SPA Single-Page Applications. Este framework adapta y amplía el HTML tradicional para servir mejor contenido dinámico a través de un data binding bidireccional que permite la sincronización automática de modelos y vistas. En este caso se utiliza angular para el control de los métodos en el código JavaScript de cada página HTML, aprovechando la facilidad del patrón de diseño que ofrecido, Modelo-Vista-Controlador. También se utiliza en el manejo de datos de las tablas en las interfaces de usurario, así como la actualización de dichas tablas.
- PostgreSQL: Es un Sistema de gestión de bases de datos relacional orientado a objetos y libre, publicado bajo la licencia PostgreSQL. Como muchos otros proyectos de código abierto, el desarrollo de PostgreSQL no es manejado por una empresa o persona, sino que es dirigido por una comunidad de desarrolladores que trabajan de forma desinteresada, altruista, libre o apoyados por organizaciones comerciales. Dicha comunidad es denominada el PGDG (*PostgreSQL Global Development Group*). En este caso este software se utiliza para el manejo la creación y el manejo de la base de datos del sistema.

## Seguridad

Finalmente en la parte de seguridad el sistema cuenta con filtros de seguridad para el acceso al sistema, esto porque el usuario deberá ingresar su identificación y contraseña para acceder al él, además solo podrán tener acceso al sistema aquellos usuarios que estén debidamente autorizados. También algunos usuarios van a contar con distintos privilegios por lo cual el sistema establece controles de acceso a diferentes funcionalidades del sistema según sea el privilegio que posea el usuario.

**Figura 3.** Estructura de la plataforma técnica





Fuente: Elaboración Propia

## Componentes del sistema

El sistema de correspondencia SITUN se encuentra dividido en varios módulos o secciones según las funcionalidades que debe desarrollar como sistema. A continuación se detalla cada uno de estas secciones:

- 1. Ingreso de Correspondencia:** Es la página principal del sistema la que se muestra al momento de que el usuario es reconocido por el sistema mediante el login. Se encarga de ingresar toda la información requerida de la correspondencia y guardarla en el sistema, para ello se tomará los siguientes datos: fecha de recibido, numero de oficio, fecha de oficio, destinatario, el remitente, el asunto a tratar, un recibido, el estado en que se encuentra, observaciones y Finalmente un espacio donde se selecciona si el documento contiene o no copia, en caso de tenerla si indica el documento en otro espacio. esto con el fin de lograr una buena funcionalidad individual en la que se pretende que el ingreso de los datos sea de manera precisa y consistente al sistema. Se podrá realizar siempre y cuando el ingreso de información sea correcta y que no existan espacios en blanco.

A continuación los tipos de datos que se ingresan en la correspondencia y las principales funciones que realizan la lógica respectiva a este proceso utilizando JavaScript y JQuery:

Tipos de datos:

- Fecha de Oficio: Date
- Fecha de Recibido: Date
- Destinatario: String
- Remitente: String
- Asunto: String
- Recibido: String
- Estado: String
- Numero de Oficio: String
- Observaciones: String
- Copia: String

### Funciones:

1. **controllerAngular**: Se encarga de manejar todas las funciones de este módulo de ingreso de correspondencia. Cumpliendo la función de controlador.
2. **limpiarCamposEnfocados**: Limpia los campos cuando se enfocan.
3. **checkCampoOficio**: Cambia el valor del campo referente al número de oficio si la opción SIN OFICIO está marcada.
4. **checkCampoCopia**: Cambia el valor del campo COPIA por la opción SIN COPIA.
5. **ingresoCorrespondencia**: Recoge los datos de los campos y realiza el envío de la petición de inserción a la base de datos.
6. **validacionCampos**: Realiza la validación de todos los espacios a ingresar, incluyendo la alarma si fuera el caso.
7. **botonCancelar**: Función de cancelación de la operación.
8. **limpiarCampos**: Limpia el valor de los campos de entrada.
9. **limpiaDivMensaje**: Limpia el div con el id=mensaje.
10. **cambioClase1**: Realiza un cambio de clase a los espacios de entrada del formulario de la clase has-error a from-group.
11. **cambioClase2**: Realiza el cambio de clase de los espacio de fecha del formulario a la clase has-error.
12. **cambioClase3**: Realiza un cambio de clase a los campos de entrada del formulario de la from-group a la clase has-error
13. **existeCorrespondencia**: Realiza la consulta si la correspondencia a ingresar ya existe.
14. **subePagina**: Sube la página en caso de error o éxito en el ingreso de Correspondencias.

Esta sección contiene otra funcionalidad la cual es el ingreso de alarmas para cumplir con una función de recordatorio en el caso de los documentos que requieren respuesta, en cuyo caso se indica la fecha de aviso y la fecha Límite. Además el sistema cuenta con una tabla en donde se muestran las alarmas próximas, en donde se pueden eliminar cuando ya se contestó el documento al cuál pertenecía la alarma.

A continuación los tipos de datos que se ingresan en la programación de una alarma y las principales funciones que realizan la lógica respectiva a este proceso utilizando JavaScript y JQuery:

### Tipos de datos:

- Fecha de Inicio: Date

➤ Fecha de Limite: Date

Funciones:

1. actualizaTablaAlarmas: Actualiza la tabla de alarmas.
2. eliminaAlarma: Elimina las alarmas de la tabla
3. programaAlarma: Programa una alarma de una correspondencia.
4. validacionCamposAlarma: Validación de los espacios de fecha límite y fecha de aviso de la alarma.

También este módulo cuenta con la posibilidad de adjuntar documentos para guardarlos junto con los archivos de configuración del sistema.

A continuación las principales funciones que realizan la lógica respectiva a este proceso utilizando JavaScript y JQuery:

datoAdjunto: Se encarga de adjuntar documentos a una correspondencia.

Finalmente cuenta con la posibilidad de realizar un enlace de la correspondencia recién ingresada con otra que se encuentra almacenada previamente en el sistema.

- 2. Búsqueda de Correspondencia:** Este módulo contiene la lógica de los procesos de búsqueda de correspondencias se encarga de realizar la búsqueda de uno o varios documentos específicos ya registrados en el sistema. Esta búsqueda podrá realizarse por varias categorías como: número de oficio, remitente, asunto, destinatario.

A continuación los tipos de datos que se ingresan para la búsqueda de correspondencia y las principales funciones que realizan la lógica respectiva a este proceso utilizando JavaScript y JQuery:

Tipos de datos:

- Destinatario: String
- Remitente: String
- Asunto: String
- Numero de Oficio: String

Funciones:

1. controllerAngular: Se encarga de manejar todas las funciones de este módulo de búsqueda de correspondencia. Cumpliendo la función de controlador.
2. buscaEnlaces: Muestra todos los enlaces relacionados con una correspondencia específica.
3. asignarInformacion: Busca la correspondencia y muestra la información de esta.
4. cargarDatos: Muestra los datos de la correspondencia en pantalla.
5. ObtenerRecibido: busca el nombre de la persona que recibió la correspondencia y lo cambia por el id de la persona.
6. cargarRecibido: Carga la información de la persona en el espacio del recibido o retorna la persona.
7. busquedaCorrespondencia: Método de Búsqueda de la correspondencia.
8. tipoBusqueda: Toma el tipo de búsqueda y regresa el sufijo correspondiente a la dirección del servidor.
9. tipoBusquedaBD: Toma el criterio de búsqueda y devuelve la columna que se debe buscar en la Base de datos.
10. limpiaTabla: limpia el div con el id de buscar.
11. limpiaCamposEnfocados: Limpia los campos cuando se enfocan.
12. limpiaDivMensaje: Limpia el div con el id=mensaje.

Este módulo también permite modificar los datos de una correspondencia.

A continuación los tipos de datos que se ingresan para editar la correspondencia y las principales funciones que realizan la lógica respectiva a este proceso utilizando JavaScript y JQuery:

Tipos de datos:

- Fecha de Oficio: Date
- Fecha de Recibido: Date
- Destinatario: String
- Remitente: String
- Asunto: String
- Recibido: String
- Estado: String
- Numero de Oficio: String
- Observaciones: String
- Copia: String

Funciones

1. `actualizarInfo`: Actualiza la información que se haya editado en pantalla.
2. `checkCampoOficio`: Cambia el valor del campo referente al número de oficio si la opción SIN OFICIO está marcada.
3. `validar`: Realiza la validación de todos los espacios a editar, incluyendo la alarma si fuera el caso.
4. `cambioClase2`: Realiza el cambio de clase de los campo de fecha del formulario a la clase has-error. La fecha debe de llevar el formato YYYY/MM/dd.
5. `validacionCamposAlarma`: Validación de los espacios de fecha límite y fecha de aviso de la alarma.
6. `actualizarCorrespondencia`: Recoge los datos de los campos y realiza la función de inserción.

Esta sección también permite modificar los datos de una alarma o agregar una en caso de que la correspondencia no tenga y el usuario desee agregarla.

A continuación los tipos de datos que se ingreso de una nueva alarma o modificación de una ya existente y las principales funciones que realizan la lógica respectiva a este proceso utilizando JavaScript y JQuery:

Tipos de datos:

- Fecha de Inicio: Date
- Fecha de Limite: Date

Funciones:

1. `validaFechas`: validación de fechas editadas.
2. `validaFechasAlarma`: validación de fechas de ajuste de alarmas.
3. `ajusteAlarma`: Método de ajuste de alarma.
4. `cargaAlarma`: carga los datos de la alarma para mostrar en pantalla.
5. `actualizarAlarma`: Recoge los datos de los campos y realiza la petición de actualización de alarma.
6. `nuevaAlarma`: Inserción de una nueva alarma a las correspondencia escogida.

3. **Usuarios**: Este módulo contiene la lógica de los procesos de ingreso y búsqueda de usuarios del sistema.

A continuación los tipos de datos que se ingresan para una agregación de un nuevo usuario al sistema, eliminar un usuario del sistema o realizar una búsqueda, en ese caso la búsqueda se realiza por Nombre, Apellido1, Apellido 2 o

Identificación. También adjunto a esto las principales funciones que realizan la lógica respectiva a este proceso utilizando JavaScript y JQuery:

Tipos de datos:

- Nombre: String
- Apellido 1: String
- Apellido 2: String
- Identificación: String
- Contraseña: String
- Administrador: checkBox

Funciones:

1. controllerAngular: Se encarga de manejar todas las funciones de este módulo de Usuarios. Cumpliendo la función de controlador.
2. ingresarInformación: Recoge los datos del usuario y realiza la petición de inserción a la base de datos.
3. validaCampos: Valida los espacios donde se ingresan los datos.
4. limpiarCampos: Limpia el valor de los campos de entrada.
5. cambioClase1: Realiza un cambio de clase a los espacios de entrada del formulario de la clase has-error a from-group.
6. cambioClase2: Realiza el cambio de clase de los espacio de fecha del formulario a la clase has-error.
7. botonCancelar: Función de cancelación de la operación.
8. actualizarInfo: Actualiza la información en la base de datos de un usuario.
9. busquedaUsuario: Método de Búsqueda.
10. eliminarUsuarios: Elimina los usuarios de la base de datos.

4. **Enlace de documentos:** Este módulo contiene la lógica de los procesos de enlace de documentos de correspondencia.

Tipos de datos:

- Destinatario: String
- Remitente: String

- Asunto: String
- Numero de Oficio: String

#### Funciones:

1. controllerAngular: Se encarga de manejar todas las funciones de este módulo de Usuarios. Cumpliendo la función de controlador.
2. busqueda: Metodo que realiza la búsqueda de la correspondencia
3. crearEnlace: Crea el enlace entre documentos seleccionados mediante la búsqueda.
4. limpiarCampos: Limpia el valor de los campos de entrada.

5. **Generar Reportes:** Este módulo contiene la lógica de los procesos de generar reportes de correspondencia.

#### Tipos de datos:

- Fecha de Inicio: Date
- Fecha de Limite: Date

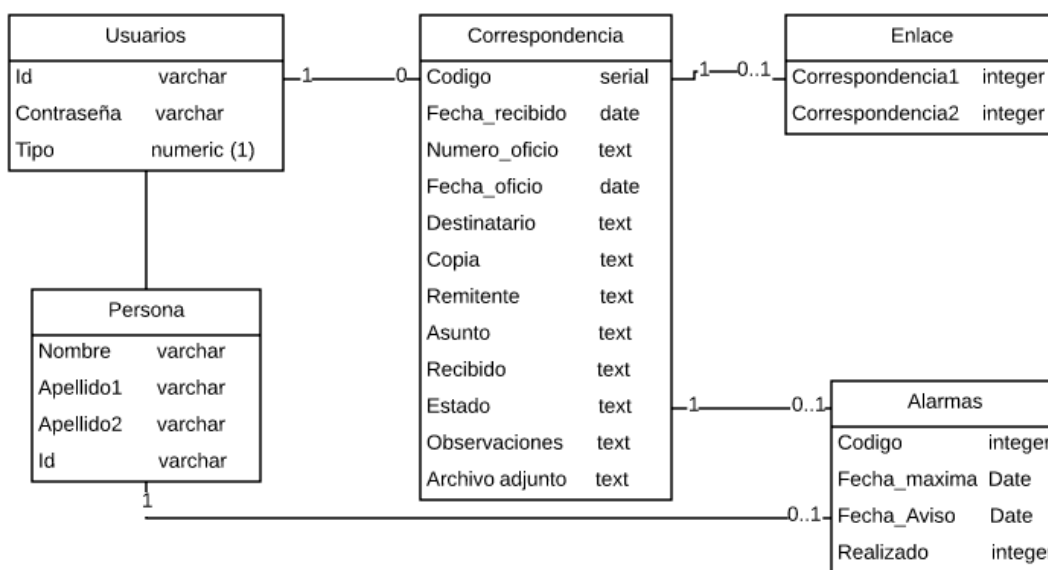
#### Funciones:

1. controllerAngular: Se encarga de manejar todas las funciones de este módulo de Generar Reportes de Correspondencias. Cumpliendo la función de controlador.
2. buscarCorrespondencias: Método de búsqueda, substraer las fechas y las envía como rango de fecha.
3. limpiarPantalla: Limpia el valor de los campos de entrada.
4. crearReporte: Método que genera el reporte de correspondencia.
5. validarEspaciosVacios: Método para validar fechas vacías.
6. validarRangoDeFechas: Método para validar rango de fechas.

## Base de datos

A continuación se presenta el modelo de base de datos utilizado, sus tablas y los tipos de datos.

**Figura 4. Ejemplo de Modelo Relacional**



**Fuente: Elaboración propia**

Como se puede observar en la figura anterior se muestra el diagrama del modelo relacional utilizado para la base de datos, en donde se puede apreciar la cardinalidad y las relaciones entre las diferentes tablas.

Tablas:

1. Tabla Persona: Esta tabla es utilizada para almacenar los datos de una persona que a su vez también es usuario del sistema.

Atributos:

- Nombre: varchar
- Apellido 1: varchar



- Apellido 2: varchar
- Id: varchar ( Llave Primaria )

2. Tabla Usuarios: Esta tabla se utiliza para almacenar los datos del usuario en este caso tipo, id y contraseña.

Atributos:

- Nombre: varchar
- Apellido 1: varchar
- Apellido 2: varchar
- Id: varchar ( Llave Foránea )

En este caso la llave foránea Id tiene como referencia el atributo Id de la tabla persona.

3. Tabla Correspondencia:

Atributos:

- Código: Serial ( Llave Primaria)
- Fecha de Recibido: Date
- Numero de Oficio: text
- Fecha de Oficio: Date
- Destinatario: text
- Copia: text
- Remitente: text
- Asunto: text
- Recibido: text ( Llave Foránea)
- Estado: text
- Observaciones: text
- Adjunto: text

En este caso la llave foránea Recibido tiene como referencia el atributo Id de la tabla persona.

4. Tabla Alarmas

Atributos:

- Código: Integer (Llave Foránea) , ( Llave Primaria Compuesta)
- Fecha Máxima: Date ( Llave Primaria Compuesta)

- Fecha Aviso: Date
- Realizado:

En este caso la llave foránea Código tiene como referencia el atributo Código de la tabla correspondencia.

La llave primaria está compuesta por dos atributos el Código y la Fecha Máxima.

## 5. Tabla Enlaces

Atributos:

- Correspondencia1: Integer (Llave Foránea) , ( Llave Primaria Compuesta)
- Correspondencia2: Integer (Llave Foránea) , ( Llave Primaria Compuesta)

En este caso la llave foránea Correspondencia1 tiene como referencia el atributo Código de la tabla correspondencia.

La llave foránea Correspondencia2 tiene como referencia el atributo Código de la tabla correspondencia.

La llave primaria está compuesta por dos atributos Correspondencia1 y Correspondencia2.

**Catálogo de Funciones Almacenadas**

Identificador	Parámetros	Versión	Detalles
Post	xn integer	1	Busca los enlaces superiores de una correspondencia
Pre	xn integer	1	Busca los enlaces inferiores de una correspondencia
Final	x integer	1	Llama las dos funciones anteriores Post y Pre y las ejecuta
Enlaces	x integer	1	Llama la función anterior Final y la ejecuta