**ACTIVIDAD I - UNIDAD I**



**Estudiante: Mario Felipe Goenaga Lozano**

**AF17202: Programación Orientada a Objetos**

**Tutor: Dr. John Carlos Arrieta Arrieta**

**Programa de Ingeniería de Software**

**Facultad de Ciencias Económicas e Ingeniería**

**Universidad de Cartagena**

**01 de Abril de 2024**

**Introducción**

El paradigma de la Programación Orientada a Objetos (POO) hace posible el desarrollo de aplicaciones caracterizado por la flexibilidad, por ello, resulta especialmente idóneo para las necesidades del entorno empresarial, en donde se hace necesario llevar a cabo procesos de estructuración, planificación, ejecución y control.

Por ejemplo, en lo que compete al ejercicio del que trata este trabajo: el desarrollo de una aplicación para el proceso de gestión de proyectos de una empresa, el cual es indispensable pues a través de proyectos es cómo estas organizaciones pueden introducir nuevas ideas de productos y/o servicios y mantenerse competitivas en un entorno altamente cambiante, el paradigma de la POO y su capacidad para modelar el mundo real, descomponiéndolo en objetos con atributos y comportamientos, es ideal.

Así tenemos que, dado que los proyectos involucran la relación entre elementos como: tareas, empleados, persona a cargo, entre otros, la Programación Orientada a Objetos permite verlos como lo que son: objetos interconectados, cuyo enlazamiento genera procesos y jerarquías, estableciendo un marco de trabajo a través del cual se proporciona un código estructurado para dar respuesta a los requerimientos que involucra la complejidad del proceso de gestión de proyectos, el cual involucra diversas etapas, sujetos implicados y una recurrente evaluación.

A través de este ejercicio práctico, será posible adentrarnos en los principios y características del POO, además de desarrollar las habilidades para el diseño de aplicaciones en un entorno que siempre busca optimizarse: el empresarial.

**Objetivos**

1. Analizar los principios de la Programación Orientada a Objetos y cómo se utilizan en el desarrollo de software propiamente.
2. Comprender y desarrollar cada uno de los requerimientos relacionados con una aplicación para facilitar los procesos asociados a la gestión de proyectos en una organización empresarial.
3. Desarrollar y potenciar habilidades en el desarrollo de software a través del paradigma de programación orientada a objetos.

**Justificación**

La justificación de la realización de este ejercicio estriba en dos aspectos: el primero es la necesidad de desarrollar y potenciar nuestras habilidades en el desarrollo de software a través de la programación orientada a objetos, una actividad compleja pero llena de aprendizajes que nos preparan, en tanto estudiantes de ingeniería de software, para elaborar en el futuro productos acordes a los requerimientos de nuestros clientes en un entorno cada vez más cambiante y competitivo. El segundo se relaciona con lo que el ejercicio nos propone en sí: una aplicación para facilitar la aplicación de proyectos. El desarrollo de este tipo de productos de software se hace enteramente necesario puesto que en las empresas se producen procesos complejos que necesitan ser optimizados cada vez, lo cual hace más sencillo la toma de decisiones, la utilización más eficiente de recursos humanos y financieros y la disminución de errores.

**Lista de Requerimientos Funcionales de una Aplicación para la Gestión de Proyectos de una Empresa**

1. Registro de proyectos:

El sistema debe permitir el registro de proyectos y datos relacionados como un nombre en clave, su denominación comercial, fecha de inicio y fecha de finalización, estado actual.

1. Registro de empleados:

El sistema requiere que se permita el registro de empleados por nombre completo, fecha de nacimiento, número de identificación o DNI, dirección, número de contacto, fecha de inicio de contratación, área de trabajo.

1. Visualizar ficha de empleados:

El sistema debe permitir a la empresa tener acceso a una ficha que muestre la información del registro de empleados.

1. Asignación de promotor:

El sistema debe permitir asignar un promotor que haga parte del grupo de empleados para los proyectos registrados teniendo en cuenta experiencia en el área de trabajo relacionada con el proyecto.

1. Dividir proyecto en tareas:

El sistema debe ayudar a dividir el proyecto en varias tareas. Estas tareas a su vez deben estar registradas según descripción, tipo, fecha de inicio estimada, fecha de inicio real, duración estimada, duración real.

1. Inscripción a tareas:

El sistema debe permitir a los empleados inscribirse a las distintas tareas en las que se descomponen los proyectos que en cada momento estén en marcha y hacer registro de tiempo dedicado por empleado a cada tarea para calcular carga de trabajo.

1. Generar documento:

El sistema debe permitir al usuario generar documentos asociados a cada tarea identificados con un código único.

1. Actualizar/revertir versión documento:

Debe permitir a ciertos usuarios actualizar o revertir versiones de documentos, así como registrar cada cambio con el nombre del responsable y la fecha de modificación.

**Lista de Clases para Proyecto:**

**“Diseño de App para Optimizar la Gestión de Procesos a Nivel Empresa”**

**Nombre Clase**: Empresa.

**Descripción**: Esta clase representa a todos los objetos de tipo Empresa, los cuales tienen relaciones con otros objetos del sistema, tales como; Empleado que es de tipo asociativo

**Propiedades de la clase**:

* id. Tipo de dato: int. Es requerido: Si
* nit. Tipo de dato: String. Es requerido: Si
* nombre. Tipo de dato: String. Es requerido: Si
* teléfono. Tipo de dato: String. Es requerido: Si
* correoElectronico. Tipo de dato: String. Es requerido: Si
* empleados. Tipo de dato: List<Empleado>. Es requerido: Si

**Métodos de la clase**:

* Empresa():
  + Objetivo: Retornar y dar valor a un objeto de tipo Empresa asignando valores por defecto a sus propiedades.
  + Ámbito: Público
  + Tipo de retorno: Empresa
  + Parámetros: Ninguno
* Empresa():
  + Objetivo: Retornar y dar valor a un objeto a de tipo Empresa asignando el valor de los parámetros a los respectivos de las propiedades del nuevo objeto a crear.
  + Ámbito: Público
  + Tipo de retorno: Empresa
  + Parámetros: (nit, nombre, teléfono, correoElectronico, empleados)
* getId():
  + Objetivo: Retornar el valor del atributo id
  + Ámbito: Público
  + Tipo de retorno: Cadena de caracteres
  + Parámetros: Ninguno
* getNit():
  + Objetivo: Retornar el valor del atributo nit
  + Ámbito: Público
  + Tipo de retorno: Cadena de caracteres
  + Parámetros: Ninguno
* setNit():
  + Objetivo: Cambiar el valor del atributo nit
  + Ámbito: Público
  + Tipo de retorno: nada
  + Parámetros: nit
* getNombre():
  + Objetivo: Retornar el valor del atributo nombre
  + Ámbito: Público
  + Tipo de retorno: Cadena de caracteres
  + Parámetros: Ninguno
* setNombre():
  + Objetivo: Cambiar el valor del atributo nombre
  + Ámbito: Público
  + Tipo de retorno: nada
  + Parámetros: nombre
* getTelefono():
  + Objetivo: Retornar el valor del atributo telefono
  + Ámbito: Público
  + Tipo de retorno: Cadena de caracteres
  + Parámetros: Ninguno
* setTelefono():
  + Objetivo: Cambiar el valor del atributo telefono
  + Ámbito: Público
  + Tipo de retorno: nada
  + Parámetros: teléfono
* getCorreoElectronico():
  + Objetivo: Retornar el valor del atributo correoElectronico
  + Ámbito: Público
  + Tipo de retorno: Cadena de caracteres
  + Parámetros: Ninguno
* setCorreoElectronico():
  + Objetivo: Cambiar el valor del atributo correoElectronico
  + Ámbito: Público
  + Tipo de retorno: nada
  + Parámetros: nombre
* getEmpleados():
  + Objetivo: Retornar el valor del atributo empleados
  + Ámbito: Público
  + Tipo de retorno: List<Empleado>
  + Parámetros: Ninguno

**Nombre Clase**: Proyecto.

**Descripción**: Esta clase representa a todos los objetos de tipo Proyecto, los cuales tienen relaciones con otros objetos del sistema, tales como; Empleado que es de tipo composición, Promotor que es de tipo asociativo y Tarea la cual también es de tipo asociativo.

**Propiedades de la clase:**

* id. Tipo de dato: int. Es requerido: Si
* nombreClave. Tipo de dato: String. Es requerido: Si
* denomComercial. Tipo de dato: String. Es requerido: Si
* fechaInicio. Tipo de dato: String. Es requerido: Si
* fechaFinalizacion. Tipo de dato: String. Es requerido: Si
* Estado. Tipo de dato: String. Es requerido: Si
* codigoProyecto. Tipo de dato: String. Es requerido: Si
* promotor. Tipo de dato: Promotor. Es requerido: Si
* tareas. Tipo de dato: List<Tarea>. Es requerido: Si

**Métodos de la clase**:

* Proyecto():
  + Objetivo: Retornar y dar valor a un objeto de tipo Proyecto asignando valores por defecto a sus propiedades.
  + Ámbito: Público
  + Tipo de retorno: Proyecto
  + Parámetros: Ninguno
* Proyecto():
  + Objetivo: Retornar y dar valor a un objeto a de tipo Proyecto asignando el valor de los parámetros a los respectivos de las propiedades del nuevo objeto a crear.
  + Ámbito: Público
  + Tipo de retorno: Proyecto
  + Parámetros: (nombreClave, denomComercial, fechaInicio, fechaFinalizacion, estado, codigoProyecto, promotor, tareas)
* getId():
  + Objetivo: Retornar el valor del atributo id
  + Ámbito: Público
  + Tipo de retorno: Cadena de caracteres
  + Parámetros: Ninguno
* getNombreClave():
  + Objetivo: Retornar el valor del atributo nombreClave
  + Ámbito: Público
  + Tipo de retorno: Cadena de caracteres
  + Parámetros: Ninguno
* setNombreClave ():
  + Objetivo: Cambiar el valor del atributo nombreClave
  + Ámbito: Público
  + Tipo de retorno: nada
  + Parámetros: nombreClave
* getDenomComercial ():
  + Objetivo: Retornar el valor del atributo denomComercial
  + Ámbito: Público
  + Tipo de retorno: Cadena de caracteres
  + Parámetros: Ninguno
* setDenomComercial ():
  + Objetivo: Cambiar el valor del atributo denomComercial
  + Ámbito: Público
  + Tipo de retorno: nada
  + Parámetros: denomComercial
* getFechaInicio ():
  + Objetivo: Retornar el valor del atributo fechaInicio
  + Ámbito: Público
  + Tipo de retorno: Cadena de caracteres
  + Parámetros: Ninguno
* setFechaInicio ():
  + Objetivo: Cambiar el valor del atributo fechaInicio
  + Ámbito: Público
  + Tipo de retorno: nada
  + Parámetros: fechaInicio
* getFechaFinalizacion ():
  + Objetivo: Retornar el valor del atributo fechaFinalizacion
  + Ámbito: Público
  + Tipo de retorno: Cadena de caracteres
  + Parámetros: Ninguno
* setFechaFinalizacion ():
  + Objetivo: Cambiar el valor del atributo fechaFinalizacion
  + Ámbito: Público
  + Tipo de retorno: nada
  + Parámetros: fechaFinalizacion
* getEstado ():
  + Objetivo: Retornar el valor del atributo estado Ámbito: Público
  + Tipo de retorno: Cadena de caracteres
  + Parámetros: Ninguno
* setEstado ():
  + Objetivo: Cambiar el valor del atributo estado Ámbito: Público
  + Tipo de retorno: nada
  + Parámetros: estado
* getCodigoProyecto ():
  + Objetivo: Retornar el valor del atributo codigoProyecto
  + Ámbito: Público
  + Tipo de retorno: Cadena de caracteres
  + Parámetros: Ninguno
* setCodigoProyecto ():
  + Objetivo: Cambiar el valor del atributo codigoProyecto
  + Ámbito: Público
  + Tipo de retorno: nada
  + Parámetros: codigoProyecto
* getPromotor ():
  + Objetivo: Retornar el valor del atributo promotor
  + Ámbito: Público
  + Tipo de retorno: Cadena de caracteres
  + Parámetros: Ninguno
* setPromotor ():
  + Objetivo: Cambiar el valor del atributo promotor
  + Ámbito: Público
  + Tipo de retorno: nada
  + Parámetros: promotor
* getTareas():
  + Objetivo: Retornar el valor del atributo tareas
  + Ámbito: Público
  + Tipo de retorno: List<Tarea>
  + Parámetros: Ninguno

**Nombre Clase**: Usuario

**Descripción**: Esta clase representa a todos los objetos de tipo Usuario, los cuales tienen relaciones con otros objetos del sistema, tales como; Empleado que es de tipo Herencia.

**Propiedades de la clase:**

* usuario. Tipo de dato: String. Es requerido: Si
* contrasena. Tipo de dato: String. Es requerido: Si
* estado. Tipo de dato: String. Es requerido: Si

**Métodos de la clase**:

* Usuario():
  + Objetivo: Retornar y dar valor a un objeto de tipo Usuario asignando valores por defecto a sus propiedades.
  + Ámbito: Público
  + Tipo de retorno: Usuario
  + Parámetros: Ninguno
* Usuario():
  + Objetivo: Retornar y dar valor a un objeto a de tipo Usuario asignando el valor de los parámetros a los respectivos de las propiedades del nuevo objeto a crear.
  + Ámbito: Público
  + Tipo de retorno: Usuario
  + Parámetros: (usuario, contrasena, estado)
* getUsuario():
  + Objetivo: Retornar el valor del atributo usuario
  + Ámbito: Público
  + Tipo de retorno: Cadena de caracteres
  + Parámetros: Ninguno
* setUsuario():
  + Objetivo: Cambiar el valor del atributo Usuario
  + Ámbito: Público
  + Tipo de retorno: nada
  + Parámetros: usuario
* getContrasena():
  + Objetivo: Retornar el valor del atributo contrasena
  + Ámbito: Público
  + Tipo de retorno: Cadena de caracteres
  + Parámetros: Ninguno
* setContrasena():
  + Objetivo: Cambiar el valor del atributo contrasena
  + Ámbito: Público
  + Tipo de retorno: nada
  + Parámetros: contrasena
* getEstado():
  + Objetivo: Retornar el valor del atributo estado
  + Ámbito: Público
  + Tipo de retorno: Cadena de caracteres
  + Parámetros: Ninguno
* setEstado ():
  + Objetivo: Cambiar el valor del atributo estado
  + Ámbito: Público
  + Tipo de retorno: nada
  + Parámetros: estado

**Nombre Clase**: Promotor

**Descripción**: Esta clase representa a todos los objetos de tipo Promotor, los cuales tienen relaciones con otros objetos del sistema, tales como; Empleado que es de tipo Herencia y Proyecto que es de tipo asociativo.

**Propiedades de la clase:**

* id. Tipo de dato: int. Es requerido: Si
* area. Tipo de dato: String. Es requerido: Si
* proyectos. Tipo de dato: Proyecto. Es requerido: Si

**Métodos de la clase**:

* Promotor():
  + Objetivo: Retornar y dar valor a un objeto de tipo Promotor asignando valores por defecto a sus propiedades.
  + Ámbito: Público
  + Tipo de retorno: Promotor
  + Parámetros: Ninguno
* Promotor():
  + Objetivo: Retornar y dar valor a un objeto a de tipo Promotor asignando el valor de los parámetros a los respectivos de las propiedades del nuevo objeto a crear.
  + Ámbito: Público
  + Tipo de retorno: Promotor
  + Parámetros: (area, proyectos)
* getId():
  + Objetivo: Retornar el valor del atributo id
  + Ámbito: Público
  + Tipo de retorno: Cadena de carácter
  + Parámetros: Ninguno
* getArea():
  + Objetivo: Retornar el valor del atributo area
  + Ámbito: Público
  + Tipo de retorno: Cadena de caracteres
  + Parámetros: Ninguno
* setArea():
  + Objetivo: Cambiar el valor del atributo area
  + Ámbito: Público
  + Tipo de retorno: nada
  + Parámetros: area
* getProyectos():
  + Objetivo: Retornar el valor del atributo proyectos
  + Ámbito: Público
  + Tipo de retorno: List<Proyecto>
  + Parámetros: Ninguno

**Nombre de clase:** FuerzaLaboral:

**Descripción:** esta clase representa una relación de muchos a muchos entre los objetos Empleado y el objeto Proyecto. Existiendo una relación de inclusión entre ellos

**Propiedades de las clases:**

* id. Tipo: int. Es requerido: si
* empleado. Tipo: Empleado. Es requerido: si
* proyecto. Tipo: Proyecto. Es requerido: si

**Métodos de la clase:**

* + FuerzaLaboral():
  + Objetivo: Retornar y dar valor a un objeto de tipo FuerzaLaboral asignando valores por defecto a sus propiedades.
* Ámbito: publico;
* Tipo de retorno: FuerzaLaboral
* Parámetros: ninguno
  + FuerzaLaboral():
* Objetivo: Inicializar y retornar un objeto FuerzaLaboral dándole un valor inicial a las propiedades con el valor de los parámetros.
* Ámbito: publico
* Tipo de retorno: FuerzaLaboral
* Parámetros: empleado, proyecto
  + getId():
* Objetivo: retornar el valor de la propiedad id de la clase Persona Física.
* Ámbito: publico
* Tipo de retorno: Numero Entero
* Parámetros: ninguno
  + getEmpleado():
* Objetivo: retornar el valor de la propiedad Empleado.
* Ámbito: publico
* Tipo de retorno: Empleado
* Parámetros: ninguno
  + getProyecto():
* Objetivo: retornar el valor de la propiedad Proyecto.
* Ámbito: publico
* Tipo de retorno: Proyecto
* Parámetros: ninguno

**Nombre de clase:** Empleado:

**Descripción:** esta clase representa a todos los empleados del sistema, cuenta con relaciones de asosacion con los objetos Empresa, Proyecto, Tarea y hereda de la clase Usuario, ademas es la clase padre de el objeto promotor.

**Propiedades de las clases:**

* id. Tipo: int. Es requerido: si
* dni. Tipo: String. Es requerido: si
* nombre. Tipo: String. Es requerido: si
* apellido. Tipo: String. Es requerido: si
* direccion. Tipo: String. Es requerido: si
* telefono. Tipo: String. Es requerido: si
* correoElectronico. Tipo: String. Es requerido: si
* fechaContratacion. Tipo: Date. Es requerido: si
* rol. Tipo: String. Es requerido: si
* tipoContrato. Tipo: String. Es requerido: si
* fechaNacimiento. Tipo: Date. Es requerido: si
* - empresa. Tipo: Empresa. Es requerido: si

**Métodos de la clase:**

* + Empleado():
* Objetivo: crear o retornar un objeto Empleado dándole valores por defecto a todas sus propiedades.
* Ámbito: publico;
* Tipo de retorno: Empleado
* Parámetros: ninguno
  + Empleado():
* Objetivo: Inicializar y retornar un objeto Empleado dándole un valor inicial a las propiedades con el valor de los parámetros.
* Ámbito: publico
* Tipo de retorno: Empleado
* Parámetros: dni, nombre, apellido, direccion, teléfono, correoElectronico, fechaContratacion, rol, tipoContrato, fechaNacimiento, empresa
  + getDni():
* Objetivo: retornar el valor de la propiedad Dni.
* Ámbito: publico
* Tipo de retorno: Cadena de Caracteres
* Parámetros: ninguno
  + SetDni():
* Objetivo: cambiar el valor de la propiedad Dni.
* Ámbito: publico
* Tipo de retorno: ninguno
* Parámetros: dni
  + getNombre():
* Objetivo: retornar el valor de la propiedad Nombre.
* Ámbito: publico
* Tipo de retorno: Cadena de Caracteres
* Parámetros: ninguno
  + SetNombre():
* Objetivo: cambiar el valor de la propiedad Nombre.
* Ámbito: publico
* Tipo de retorno: ninguno
* Parámetros: nombre
* o
  + getApellido():
* Objetivo: retornar el valor de la propiedad Apellido.
* Ámbito: publico
* Tipo de retorno: Cadena de Caracteres
* Parámetros: ninguno
  + SetApellido():
* Objetivo: cambiar el valor de la propiedad Apellido.
* Ámbito: publico
* Tipo de retorno: ninguno
* Parámetros: apellido
  + getDireccion():
* Objetivo: retornar el valor de la propiedad Direccion.
* Ámbito: publico
* Tipo de retorno: Cadena de Caracteres
* Parámetros: ninguno
  + SetDireccion():
* Objetivo: cambiar el valor de la propiedad Direccion.
* Ámbito: publico
* Tipo de retorno: ninguno
* Parámetros: direccion
  + getTelefono():
* Objetivo: retornar el valor de la propiedad Telefono.
* Ámbito: publico
* Tipo de retorno: Cadena de Caracteres
* Parámetros: ninguno
  + SetTelefono():
* Objetivo: cambiar el valor de la propiedad Telefono.
* Ámbito: publico
* Tipo de retorno: ninguno
* Parámetros: teléfono
  + getCorreoElectronico():
* Objetivo: retornar el valor de la propiedad Correo Electrónico.
* Ámbito: publico
* Tipo de retorno: Cadena de Caracteres
* Parámetros: ninguno
  + SetCorreoElectronico ():
* Objetivo: cambiar el valor de la propiedad Correo Electrónico.
* Ámbito: publico
* Tipo de retorno: ninguno
* Parámetros: correoElectronico
  + getFechaContratacion():
* Objetivo: retorna el valor de la propiedad FechaContratacion.
* Ámbito: publico
* Tipo de retorno: Date
* Parámetros: Ninguno
  + setFechaContratacion():
* Objetivo: retorna el valor de la propiedad FechaContratacion.
* Ámbito: publico
* Tipo de retorno: Ninguna
* Parámetros: fechaContratacion
  + getRol():
* Objetivo: retornar el valor de la propiedad Rol.
* Ámbito: publico
* Tipo de retorno: Cadena de Caracteres
* Parámetros: ninguno
  + SetRol():
* Objetivo: cambiar el valor de la propiedad Rol.
* Ámbito: publico
* Tipo de retorno: ninguno
* Parámetros: rol
  + getTipoContrato():
* Objetivo: retornar el valor de la propieda Tipo contrato.
* Ámbito: publico
* Tipo de retorno: Cadena de Caracteres
* Parámetros: ninguno
  + setTipoContrato():
* Objetivo: cambiar el valor de la propiedad Tipo Contrato.
* Ámbito: publico
* Tipo de retorno: ninguno
* Parámetros: tipoContrato
  + getFechaNacimiento ():
* Objetivo: retorna el valor de la propiedad FechaNacimiento.
* Ámbito: publico
* Tipo de retorno: Date
* Parámetros: Ninguno
  + setFechaNacimiento():
* Objetivo: retorna el valor de la propiedad FechaNacimiento.
* Ámbito: publico
* Tipo de retorno: Ninguna
* Parámetros: fechaNacimiento
  + getEmpresa():
* Objetivo: obtener el valor de la propiedad Empresa.
* Ámbito: publico
* Tipo de retorno: Empresa
* Parámetros: ninguno

**Nombre de clase:** RutaTrabajo:

**Descripción:** esta clase representa una relación de muchos a muchos entre los objetos Empleado y el objeto Tarea. Existiendo una relación de inclusión entre ellos

**Propiedades de las clases:**

* id. Tipo: int. Es requerido: si
* empleado. Tipo: Empleado. Es requerido: si
* tarea. Tipo: Tarea. Es requerido: si

**Métodos de la clase:**

* RutaTrabajo ():
* Objetivo: crear o retornar un objeto RutaTrabajo dándole valores por defecto a todas sus propiedades.
* Ámbito: publico;
* Tipo de retorno: RutaTrabajo
* Parámetros: ninguno
* RutaTrabajo ():
* Objetivo: Inicializar y retornar un objeto RutaTrabajo dándole un valor inicial a las propiedades con el valor de los parámetros.
* Ámbito: publico
* Tipo de retorno: RutaTrabajo
* Parámetros: empleado, tarea
  + getId():
* Objetivo: retornar el valor de la propiedad id de la clase Persona Física.
* Ámbito: publico
* Tipo de retorno: Numero Entero
* Parámetros: ninguno
* getEmpleado():
* Objetivo: retornar el valor de la propiedad Empleado.
* Ámbito: publico
* Tipo de retorno: Empleado
* Parámetros: ninguno
* getTarea():
* Objetivo: retornar el valor de la propiedad Tarea.
* Ámbito: publico
* Tipo de retorno: Tarea
* Parámetros: ninguno

**Nombre de clase:** Tarea:

**Descripción:** Esta clase representa a todos los objetos de tipo Tarea, y cuentan con una relación de composición con los objetos de tipo Proyecto, además tiene una relación de asociación con los objetos Empleados.

**Propiedades de las clases:**

* id. Tipo: int. Es requerido: si
* descripcion. Tipo: String. Es requerido: si
* tipo. Tipo: String. Es requerido: si
* fechaInicioEstimada. Tipo: Date. Es requerido: si
* fechaInicioReal. Tipo: Date. Es requerido: si
* duracionEstimada. Tipo: String. Es requerido: si
* duracionReal. Tipo: String. Es requerido: si
* empleado. Tipo: String. Es requerido: si
* documento. Tipo: String. Es requerido: si
* prioridad. Tipo: String. Es requerido: si
* proyecto. Tipo: Proyecto. Es requerido: si
* documentos. Tipo: List<Documentos>. Es requerido: si

**Métodos de la clase:**

* Tarea():
* Objetivo: crear o retornar un objeto tarea dándole valores por defecto a todas sus propiedades.
* Ámbito: publico;
* Tipo de retorno: Tarea
* Parámetros: ninguno
* Tarea():
* Objetivo: Inicializar y retornar un objeto Tarea dándole un valor inicial a las propiedades con el valor de los parámetros.
* Ámbito: publico
* Tipo de retorno: Tarea
* Parámetros: descripción, tipo, fechaInicioEstimada, fechaInicioReal, duracionEstimada, duracionReal, empleado, documento, prioridad, proyecto, documentos
* getId():
* Objetivo: retornar el valor de la propiedad id de la clase Persona Física.
* Ámbito: publico
* Tipo de retorno: Numero Entero
* Parámetros: ninguno
* getDescripcion():
* Objetivo: retornar el valor de la propiedad Descripción.
* Ámbito: publico
* Tipo de retorno: Cadena de Caracteres
* Parámetros: ninguno
* SetDescripcion():
* Objetivo: cambiar el valor de la propiedad descripción.
* Ámbito: publico
* Tipo de retorno: ninguno
* Parámetros: descripción
* getTipo():
* Objetivo: retornar el valor de la propiedad Tipo.
* Ámbito: publico
* Tipo de retorno: Cadena de Caracteres
* Parámetros: ninguno
* SetTipo():
* Objetivo: cambiar el valor de la propiedad Tipo.
* Ámbito: publico
* Tipo de retorno: ninguno
* Parámetros: tipo

• getFechaInicioEstimada():

* Objetivo: retorna el valor de la propiedad FechaInicioEstimada.
* Ámbito: publico
* Tipo de retorno: Date
* Parámetros: Ninguno
* setFechaInicioEstimada():
* Objetivo: retorna el valor de la propiedad FechaInicioEstimada.
* Ámbito: publico
* Tipo de retorno: Ninguna
* Parámetros: fechaInicioEstimada
* getFechaInicioReal():
* Objetivo: retorna el valor de la propiedad FechaInicioReal.
* Ámbito: publico
* Tipo de retorno: Date
* Parámetros: Ninguno
* setFechaInicioReal():
* Objetivo: retorna el valor de la propiedad FechaInicioReal.
* Ámbito: publico
* Tipo de retorno: Ninguna
* Parámetros: fechaInicioReal
* getDuracionEstimada():
* Objetivo: retornar el valor de la propiedad Duración Estimada.
* Ámbito: publico
* Tipo de retorno: Cadena de Caracteres
* Parámetros: ninguno
* SetDuracionEstimada():
* Objetivo: cambiar el valor de la propiedad Duración Estimada.
* Ámbito: publico
* Tipo de retorno: ninguno
* Parámetros: duracionEstimada
* getDuracionReal():
* Objetivo: retornar el valor de la propiedad Duración Real.
* Ámbito: publico
* Tipo de retorno: Cadena de Caracteres
* Parámetros: ninguno
* SetDuracionReal():
* Objetivo: cambiar el valor de la propiedad Duración Real.
* Ámbito: publico
* Tipo de retorno: ninguno
* Parámetros: duracionReal
* getEmpleado():
* Objetivo: retornar el valor de la propiedad Empleado.
* Ámbito: publico
* Tipo de retorno: Cadena de Caracteres
* Parámetros: ninguno
* setEmpleado():
* Objetivo: cambiar el valor de la propiedad Empleado.
* Ámbito: publico
* Tipo de retorno: ninguno
* Parámetros: empleado
* getDocumento():
* Objetivo: retornar el valor de la propiedad Documento.
* Ámbito: publico
* Tipo de retorno: Cadena de Caracteres
* Parámetros: ninguno
* SetDocumento():
* Objetivo: cambiar el valor de la propiedad Documento.
* Ámbito: publico
* Tipo de retorno: ninguno
* Parámetros: documento
* getPrioridad():
* Objetivo: retornar el valor de la propiedad Prioridad.
* Ámbito: publico
* Tipo de retorno: Cadena de Caracteres
* Parámetros: ninguno
* SetPrioridad():
* Objetivo: cambiar el valor de la propiedad Prioridad.
* Ámbito: publico
* Tipo de retorno: ninguno
* Parámetros: prioridad
* getProyecto():
* Objetivo: obtener el valor de la propiedad Proyecto.
* Ámbito: publico
* Tipo de retorno: Proyecto
* Parámetros: ninguno
* getDocumentos():
* Objetivo: retornar el valor de la propiedad Documentos.
* Ámbito: publico
* Tipo de retorno: List<Documento>
* Parámetros: ninguno

**Nombre de clase:** Documento:

**Descripción:** Esta clase representa a todos los objetos de tipo

Documento, y cuentan con una relación de composición con los

objetos de tipo Tarea.

**Propiedades de las clases:**

* id. Tipo: int. Es requerido: si
* proyecto. Tipo: String. Es requerido: si
* codigoDocumento. Tipo: String. Es requerido: si
* fechaDocumento. Tipo: Date. Es requerido: si
* descripcion. Tipo: String. Es requerido: si
* tarea. Tipo: Tarea. Es requerido: si
* versiones. Tipo: List<Version>. Es requerido: si

**Métodos de la clase:**

• Documento():

* Objetivo: crear o retornar un objeto Documento dándole valores por defecto a todas sus propiedades.
* Ámbito: publico;
* Tipo de retorno: Documento
* Parámetros: ninguno

• Documento():

* Objetivo: Inicializar y retornar un objeto Documento dándole un valor inicial a las propiedades con el valor de los parámetros.
* Ámbito: publico
* Tipo de retorno: Documento
* Parámetros:

• getId():

* Objetivo: retornar el valor de la propiedad id de la clase Persona Física.
* Ámbito: publico
* Tipo de retorno: Numero Entero
* Parámetros: ninguno

• getProyecto():

* Objetivo: retornar el valor de la propiedad Proyecto.
* Ámbito: publico
* Tipo de retorno: Cadena de Caracteres
* Parámetros: ninguno

• SetProyecto():

* Objetivo: cambiar el valor de la propiedad Proyecto.
* Ámbito: publico
* Tipo de retorno: ninguno
* Parámetros: proyecto

• getCodigoDocumento():

* Objetivo: retornar el valor de la propiedad Código Documento.
* Ámbito: publico
* Tipo de retorno: Cadena de Caracteres
* Parámetros: ninguno

• SetCodigoDocumento():

* Objetivo: cambiar el valor de la propiedad Código Documento.
* Ámbito: publico
* Tipo de retorno: ninguno
* Parámetros: codigoDocumento

• getFechaDocumento():

* Objetivo: retorna el valor de la propiedad Fecha Documento.
* Ámbito: publico
* Tipo de retorno: Date
* Parámetros: Ninguno

• setFechaDocumento():

* Objetivo: retorna el valor de la propiedad Fecha Documento.
* Ámbito: publico
* Tipo de retorno: Ninguna
* Parámetros: fechaDocumento

• getDescripcion():

* Objetivo: retornar el valor de la propiedad Descripción.
* Ámbito: publico
* Tipo de retorno: Cadena de Caracteres
* Parámetros: ninguno

• SetDescripcion():

* Objetivo: cambiar el valor de la propiedad descripción.
* Ámbito: publico
* Tipo de retorno: ninguno
* Parámetros: descripción

• getTarea():

* Objetivo: obtener el valor de la propiedad Tarea.
* Ámbito: publico
* Tipo de retorno: Tarea
* Parámetros: ninguno

• getVersiones():

* Objetivo: retornar el valor de la propiedad Versiones.
* Ámbito: publico
* Tipo de retorno: List<Versiones>
* Parámetros: ninguno

**Nombre de clase:** Version:

**Descripción:** Esta clase representa a todos los objetos de tipo Version, y cuentan con una relación de composición con los objetos de tipo Documento.

**Propiedades de las clases:**

* id. Tipo: int. Es requerido: si
* fecha. Tipo: Date. Es requerido: si
* descripcion. Tipo: String. Es requerido: si
* empleado. Tipo: String. Es requerido: si
* documento. Tipo: Documento. Es requerido: si

**Métodos de la clase:**

• Version():

* Objetivo: crear o retornar un objeto Version dándole valores por defecto a todas sus propiedades.
* Ámbito: publico;
* Tipo de retorno: Version
* Parámetros: ninguno

• Version():

* Objetivo: Inicializar y retornar un objeto Version dándole un valor inicial a las propiedades con el valor de los parámetros.
* Ámbito: publico
* Tipo de retorno: Version
* Parámetros:
  + getId():
* Objetivo: retornar el valor de la propiedad id de la clase Persona Física.
* Ámbito: publico
* Tipo de retorno: Numero Entero
* Parámetros: ninguno

• getDescripcion():

* Objetivo: retornar el valor de la propiedad Descripción.
* Ámbito: publico
* Tipo de retorno: Cadena de Caracteres
* Parámetros: ninguno

• SetDescripcion():

* Objetivo: cambiar el valor de la propiedad descripción.
* Ámbito: publico
* Tipo de retorno: ninguno
* Parámetros: descripción

• getEmpleado():

* Objetivo: retornar el valor de la propieda Empleado.
* Ámbito: publico
* Tipo de retorno: Cadena de Caracteres
* Parámetros: ninguno

• setEmpleado():

* Objetivo: cambiar el valor de la propiedad Empleado.
* Ámbito: publico
* Tipo de retorno: ninguno
* Parámetros: empleado
  + getFecha():
* Objetivo: retorna el valor de la propiedad Fecha.
* Ámbito: publico
* Tipo de retorno: Date
* Parámetros: Ninguno

• setFecha ():

* Objetivo: retorna el valor de la propiedad Fecha.
* Ámbito: publico
* Tipo de retorno: Ninguna
* Parámetros: fecha

• getDocumento():

* Objetivo: obtener el valor de la propiedad Documento.
* Ámbito: publico
* Tipo de retorno: Documento
* Parámetros: ninguno

Relaciones entre clases

En este apartamos mostramos las relaciones que existen entre las diferentes clases

Empleado "1..\*" -- "1..\*" Tarea : Realiza

(Empleado, Tarea) .. RutaTrabajo

Empresa "1" -- "\*" Empleado : Trabaja

Promotor "1" \*-- "\*" Proyecto : Promueve

Documento "1" \*-- "\*" Version : Tiene

Empleado "1..\*" -- "1..\*" Proyecto : Asignado

(Empleado, Proyecto) .. FuerzaLaboral

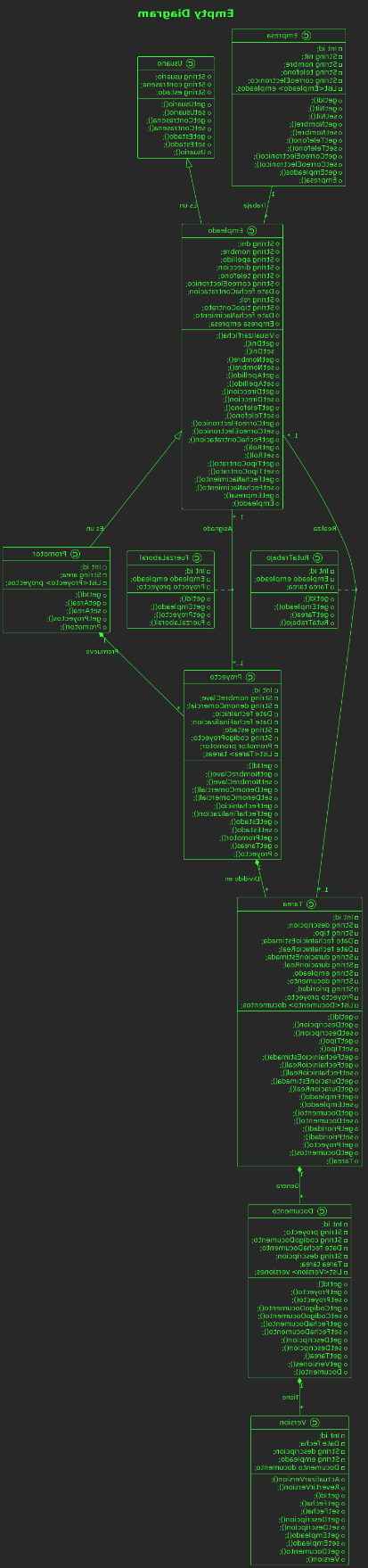
Empleado <|-- Promotor : Es un

Usuario <|-- Empleado : Es un

Tarea "1" \*-- "\*" Documento : Genera

Proyecto "1" \*-- "\*" Tarea : Dividido en

Diagrama de clases



Del diagrama al Codigo

@startuml

| @startuml !theme crt-green   title Empty Diagram  Class Empresa{ -Int id; -String nit; -String nombre; -String telefono; -String correoElectronico; -List<Empleado> empleados; +getId(); +getNit(); +setNit(); +getNombre(); +setNombre(); +getTelefono(); +setTelefono(); +getCorreoElectronico(); +setCorreoElectronico(); +getEmpleados(); +Empresa(); }  Class Proyecto{ -Int id; -String nombreClave; -String denomComercial; -Date fechaInicio; -Date fechaFinalizacion; -String estado; -String codigoProyecto; -Promotor promotor; -List<Tarea> tareas; +getId(); +getNombreClave(); +setNombreClave(); +getDenomComercial(); +setDenomComercial(); +getFechaInicio(); +getFechaFinalizacion(); +getEstado(); +setEstado(); +getPromotor(); +getTareas(); +Proyecto(); }  Class Usuario{ #String usuario; #String contrasena; #String estado; +getUsuario(); +setUsuario(); +getContrasena(); +setContrasena(); +getEstado(); +setEstado(); +Usuario(); }  Class Promotor{ -Int id; -String area; -List<Proyecto> proyectos; +getId(); +getArea(); +setArea(); +getProyectos(); +Promotor(); }  Class Tarea{ -Int id; -String descripcion; -String tipo; -Date fechaInicioEstimada; -Date fechaInicioReal; -String duracionEstimada; -String duracionReal; -String empleado; -String documento; -String prioridad; -Proyecto proyecto; -List<Documento> documentos; +getId(); +getDescripcion(); +setDescripcion(); +getTipo(); +setTipo(); +getFechaInicioEstimada(); +getFechaInicioReal(); +setFechaInicioReal(); +getDuracionEstimada(); +getDuracionReal(); +getEmpleado(); +setEmpleado(); +getDocumento(); +setDocumento(); +getPrioridad(); +setPrioridad(); +getProyecto(); +getDocumentos(); +Tarea(); }  Class Empleado{ #String dni; #String nombre; #String apellido; #String direccion; #String telefono; #String correoElectronico; #Date fechaContratacion; #String rol; #String tipoContrato; #Date fechaNacimiento; #Empresa empresa; +VisualizarFicha(); +getDni(); setDni(); +getNombre(); +setNombre(); +getApellido(); +setApellido(); +getDireccion(); +setDireccion(); +getTelefono(); +setTelefono(); +getCorreoElectronico(); +setCorreoElectronico(); +getFechaContratacion(); +getRol(); +setRol(); +getTipoContrato(); +setTipoContrato(); +getFechaNacimiento(); +setFechaNacimiento(); +getEmpresa(); +Empleado(); }  Class Documento{ -Int id; -String proyecto; -String codigoDocumento; -Date fechaDocumento; -String descripcion; -Tarea tarea; -List<Version> versiones; +getId(); +getProyecto(); +setProyecto(); +getCodigoDocumento(); +setCodigoDocumento(); +getFechaDocumento(); +setFechaDocumento(); +getDescripcion(); +setDescripcion(); +getTarea(); +getVersiones(); +Documento(); }  Class Version{ -Int id; -Date fecha; -String descripcion; -String empleado; -Documento documento; +ActualizarVersion(); +RevertirVersion(); +getId(); +getFecha(); +setFecha(); +getDescripcion(); +setDescripcion(); +getEmpleado(); +setEmpleado(); +getDocumento(); +Version(); }  Class RutaTrabajo{ -Int id; -Empleado empleado; -Tarea tarea; +getId(); +getEmpleado(); +getTarea(); +RutaTrabajo(); }  Class FuerzaLaboral{ -Int id; -Empleado empleado; -Proyecto proyecto; +getId(); +getEmpleado(); +getProyecto(); +FuerzaLaboral(); }  Empleado "1..\*" -- "1..\*" Tarea : Realiza (Empleado, Tarea) .. RutaTrabajo Empresa "1" -- "\*" Empleado : Trabaja Promotor "1" \*-- "\*" Proyecto : Promueve Documento "1" \*-- "\*" Version : Tiene Empleado "1..\*" -- "1..\*" Proyecto : Asignado (Empleado, Proyecto) .. FuerzaLaboral Empleado <|-- Promotor : Es un Usuario <|-- Empleado : Es un Tarea "1" \*-- "\*" Documento : Genera Proyecto "1" \*-- "\*" Tarea : Dividido en    @enduml |
| --- |

Bibliografía

<https://www.youtube.com/watch?v=66LhPhANswM&list=PLfB6iylhfbYuuhiSnd_aLnWzei-7o3dLm&index=7&t=2s>

<https://www.youtube.com/watch?v=aKeMmGLVQb4&list=PLfB6iylhfbYtmfgYtxlRuGGzNR6fT-MUV&index=8>