PRACTICA 2

PROGRAMACIÓN ORIENTADA A OBJETOS

Integrante:

José Santiago Pérez Piedrahita

Miguel Angel Restrepo Tangarife

Daniel Ceballos Monsalve

Carlos Daniel Urresty Ascuntar

Profesor

Jaime Alberto Guzmán Luna

Universidad Nacional de Colombia sede Medellín

Facultad de Minas

Medellín, Colombia

Junio, 2022

Descripción General de la Solución

La **Gestión de las Relaciones con Clientes (CRM)**, como el término es conocido en español, vamás alláde una plataforma o un software: es todo el proceso utilizado por startups, pequeñas y grandes empresas para **administrar y analizar las interacciones con clientes**, anticipar necesidades y deseos, optimizar larentabilidad, aumentar las ventas y personalizar campañas de captación de nuevos clientes.

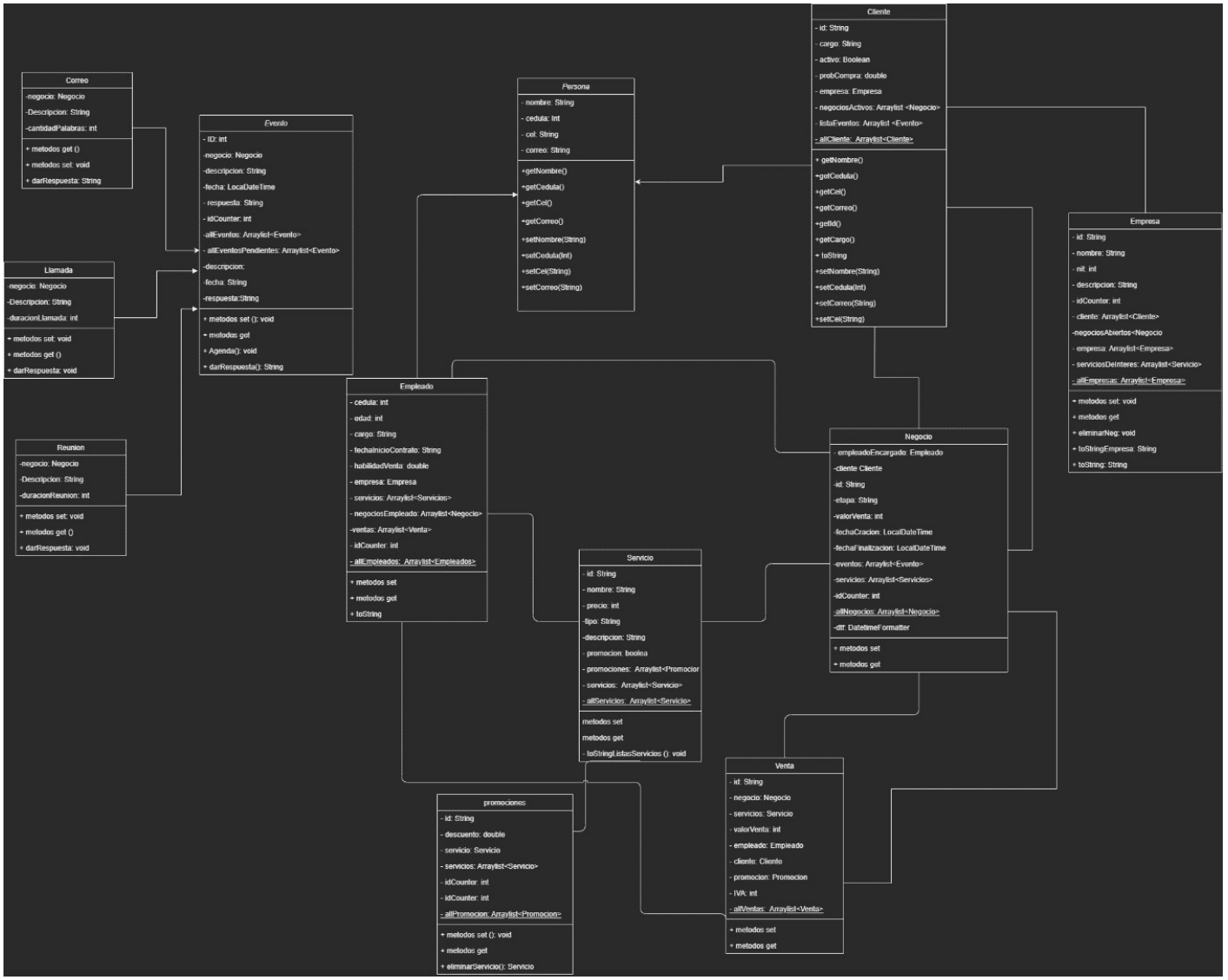
El software de CRM registra la información de contacto de los clientes, tales como el correo electrónico, el teléfono, el sitio web, el perfil de redes sociales, etc. También puede introducir otra información, como las noticias recientes sobre la actividad de la empresa, además de almacenar detalles tales como las preferencias personales de un cliente respecto a las comunicaciones Estamos en la era del cliente, de la transformación digital, de las nuevas tecnologías. En ese escenario, **la relación también evoluciona**, llevando a un nuevo concepto también conocido como **experiencia del cliente**. El concepto de CRM implica estar centrado en el cliente. Es estrategia, esun proceso, es herramienta y tecnología.

- **CRM como tecnología:** este es un producto de tecnología, a menudo en la nube, que los equipos utilizan para registrar y analizar las interacciones entre la empresa y los usuarios, y generar informessobre ellas. A esto también se le llama una solución o un sistema de CRM.
- **CRM como estrategia:** se trata de una filosofía de empresa sobre cómo deben gestionarse las relaciones con los clientes potenciales.
- **CRM como proceso:** piense en este concepto como un sistema que una empresa adopta paracuidar y gestionar las relaciones.

Nuestro proyecto: El proyecto está orientado a la creación de un sistema con el cual el usuario (vendedor) podrá almacenar los posibles clientes, empresas, incrementar el *engagement* con los anterioresa través del envío de correos, llamadas y reuniones.

Con el propósito de acotar el alcance, se decidió realizar la función de intermediario entre algún vendedor y las empresas que se encuentran en el sistema, buscando el mejor vendedor / comprador en cada caso. Para esto, identificamos clases que serían claves dentro de nuestro proyecto; un programa que busque ofrecer servicios a clientes a través de eventos dentro de un negocio, que lleve el registro de los servicios, negocios, clientes, empleados, eventos, entre otras clases que serían de vital importancia para el proyecto.

Descripción del Diseño Estático del Sistema



Adicionalmente, puede encontrar el link del proyecto de Github aquí.

Descripcion de la Implemtentacion de las Caracteristicas Orientada a Objetos

Package personas: este paquete sería el encargado de contener las clases de los actores principales del proyecto, dentro de esta se encuentran las clases Persona, Cliente, Empleado.

- **Clase Persona:** Clase abstracta abtrascta de las clases Cliente y Empleado.
- **Clase Cliente:** Esta clase se encargaria de representar el papel que toma un cliente dentro de la empresa, la cual contendría atributos de identificacion, una lista de interacciones que se han tenido con este cliente, negocios vigentes y tambien, un atributo llamado probCompra, un numero random que se asgina en su contructor que indicaría la probabilidad que tendría este cliente para comprar servicios que se le ofrecerian mas tarde.
- **Clase Empleado:** Clase hija de Persona que será la encargada de mantener contacto con el cliente a traves de una clase negocio, esta clase asignara un atributo habilidad Venta, el cual será un numero random el cual le servirá al empleado para concretar negocios. En caso de que un cliente tenga asociada una empresa esta clase se hara instanciar.

Package Evento: este paquete sería el encargado de contener las clases de engagements, que almacena la clases Correo, Llamada y Reunion

- **Clase Evento:** esta clase evento será la clase padre de las clases, reunion, correo y llamada, esta clase es importante dentro del proyecto ya que es la que se encarga de que cuando se cree un evento, se programe una reunion, se envie un correo o se realice una llamada de una respuesta, esto se hace mediante un metodo publico estatico de la clase evento llamado darRespuesta, el cual con las variables de la habilidad del empleado para vender productos y la probabilidad del cliente de comprar un servicio, mas la probabilidad que tiene cada reunion, correo o llamada se suman para dar una probabilidad, en base a este calculo, si la probabildad es muy baja, la respueta será que el cliente se da de baja en el negocio, entonces la estapa de este cambia a cerrado, si la probabilidad es media, entonces da como respuesta que el cliente no acepta pero se muestra interesado, a lo que podemos responder programando mas eventos, y si la probabilidad es muy alta, entonces el negocio avanza en la etapa

Clase Venta: esta clase se creará en cargo de que la negociacion allá finalizado y su valor haya sido positivo, los atributos de esta clase contendran valores de negocio, como servicios, precio final, cliente, empleado encargado, etc.

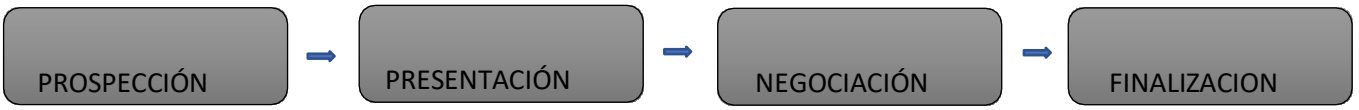
Clase Promoción: esta clase contendra tendra un valor el cual sera el que dará el descuento a un servicio asociado a esta.

Clase Registro: Servirá como constructor de clases, en caso de que se use ligadura dinamica esta registrara de manera correcta.

Clase Negocio: Esta clase es una de las mas importantes en nuestro proyecto ya que se encarga de contener asociar casi todas las clases del proyecto, pues en esta clase se contendrá los servicios que serán negociados, su valor, el empleado a cargo del negocio, el cliente a quien se le venderá los productos, la etapa en la que se encuentra el negocio, entre otras.

La clase negocio contendrá un Arraylist de Instancias de tipo Evento, los cuales según los resultados de esos eventos puede que la etapa del negocio cambia y así avance su negociacion o simplemente se cierre y no se venda nada.

- **Etapas del negocio:**



1. ***PROSPECCIÓN****:* Esta etapa se utiliza para hacer un estudio de los posibles clientes a los que les podríamos vender nuestros productos

   En nuestro programa se seleccionará un trabajador para hacer un negocio, y que productos queremos vender, cuando se abre el negocio (se crea el objeto negocio) , este tiene el atributo etapa como "prospección", con ligadura dinámica crearemos un objeto de tipo negocio que solo le asignará los valores a:

   - *ID*
   - *empleadoEncargado*
   - *etapa*
   - *fechaCreacion*
   - *servicios*

   Luego se mostrará un listado enumerado de los posibles clientes, estos clientes habrán tenido nuestros servicios en una lista de su clase llamada serviciosDeInteres, por consola se seleccionará un cliente.

2. ***PRESENTACIÓN****: En* esta etapa se el empleado intentará obtener contacto con el cliente, haciendo llamadas, enviando correos, agendando citas. Para saber si el cliente esta interesado en llegar a unacuerdo, las respuestas pueden ser:

   - Si: Entonces, la etapa del negocio cambia de Presentación a Negociación, se agreganlos servicios requeridos por el cliente a la lista servicios del objeto Negocio asociado;

- No: Entonces la etapa del negocio cambia a cerrado y no se vende (no se genera objetoVenta asociado al negocio.
- Le interesa: En este caso se deben realizar más intentos de contactos, la etapa se mantendrá

En este caso es importante saber que las respuestas se darán de acuerdo a porcentajes, como el cliente tiene servicios en los que esta interesados, entonces las probabilidades de que responda si serán del 30%, 60% si responde que le interesa y 20% si responde no.

3. ***NEGOCIACIÓN****:* Esta etapa se usará para negociar con el cliente, se dará el valor de la venta de losproductos al cliente y este tendrá tres posibles respuestas:
    - Compra: En este caso el estado de la negociación se cambia a finalización, el valor booleano venta se cambia a True y se agrega el valor final de la venta al atributo valorVenta.
    - No compra, pero sigue interesado: En este caso se repite el proceso, pero se agrega la opción de que el empleado puede hacer una promoción lo que aumenta el probabilidad de compra del cliente. (definir reglas de promociones)
    - No compra: en este caso el cliente sale de la negociación y esta pasa directamente a laetapa cierre.

4. ***FINALIZACION****:* Si la negociación tiene Boolean venta en False entonces solo se agrega la fechade finalización, si es True, entonces se crea un objeto venta con los datos de negocio. En ambos casosla etapa se cambia a Cerrado.

    Estas clases claves serían:

    - **Clase persona:** clase abstracta
    - **Empleado:** Clase cuyas instancias se encargarian de mantener contacto con los clientesde la empresa
    - **Empleado:** Esta clase sería a quien se ofrecerían serivicios

    Esta etapa se utiliza para hacer un estudio de los posibles clientes a los que les podríamos vender nuestros productos

    En nuestro programa se seleccionará un trabajador para hacer un negocio, y que productos queremos vender, cuando se abre el negocio (se crea el objeto negocio) , este tiene el atributo etapa como “prospección”, con ligadura dinámica crearemos un objeto de tipo negocio que solo le asignará los valores a:

    - *ID*
    - *empleadoEncargado*
    - *etapa*
    - *fechaCreacion*
    - *servicios*

    Luego, se mostrará un listado enumerado de los posibles clientes, estos clientes habrán tenido nuestros servicios en una lista de su clase llamada serviciosDeInteres, por consola se seleccionará un cliente.

Funcionalidades:

1. darRespuesta(): En las clases heredadas de Evento (Correo, Llamada, Reunion). Esta funcion calcula, a partir de la probabilidad de Capacidad de venta del vendedor, la probabilidad de compra del cliente, y del resultado del evento; si el cliente sigue interesado en el producto. Como se menciono anteriormente, estafuncion se encuentra definida en las clases Correo, Llamada y Reunion.
2. definirEtapa(): Esta funcion calcula en que etapa esta el negocio. Toma como entrada, la cantidad de eventos que han ocurrido en un negocio, y define si el negocio debe de avanzar a la siguiente etapa odebe finalizarse, según los resultados de los mismos. Esta funcion se encuentra en la clase Negocio.
3. Clase Promocion: Esta clase permite realizar una promocion o descuento al precio de un servicio, duranteel proceso de venta, en algun negocio.

4. Seguimiento de un Negocio. Para crear un Negocio, se debe tener un cliente/empresa y un empleado. Esto, con el fin de poder hacer seguimiento al negocio, tener claro cuales son las partes, seguir el procesode venta a traves de los eventos, etc.

5. Seguimiento de cada una de las instancias creadas según clase, a traves de un ID creado a partir de lacantidad de instancias creadas. Esto, con el fin de realizar seguimientos y limpieza de los datos usadospor el usuario. Adicionalmente, muchas funciones creadas como toString, el cual se encuentra en todaslas clases creadas, son funciones que ayudan a la interaccion del usuario con el sistema.

Requisitos Implementados

- ✓ **Clases Abstracta (1) y Métodos Abstractos (1):** La clase Persona es una clase Abstracta, lacualposee diferentes métodos abstractos.
- ✓ **Interfaces (1) diferentes a los utilizados para serializar los objetos en el punto de la persistencia. Deberá ser propio del dominio a implementar:** La clase Tiempo es una Interfaz, lacual define los métodos que deben ser implementados en la clase Persona
- ✓ **Herencia (1):** La clase Evento, es una clase que hereda a las clases Correo, Llamada, Reunión.Esta clase hereda los métodos y atributos que deben tener las otras clases
- ✓ **Ligadura dinámica (2) asociadas al modelo lógico de la aplicación:** La clase Negocio usaligadura dinámica, para decidir el método a usar en según el tipo de Evento.
- ✓ **Atributos de clase (1) y métodos de clase (1):** La mayoría de clases tienen atributos de clase, con loscuales se almacena las instancias creadas según clase. Por lo tanto, se crean los métodos de clase para acceder a ellos.
- ✓ **Uso de constante (1 caso):** En la clase Tiempo, la cual es una interfaz, posee atributos, los cualesson constantes por defectos:
- ✓ **Encapsulamiento (private, protected y public):** Usado en todos los atributos y métodos de cada una delas clases.
- ✓ **Los siguientes conceptos asociados a la POO:**
  - **Sobrecarga de métodos (1 casos mínimo) y constructores (2 casos mínimo)**
  - **Manejo de referencias this para desambiguar y this() entre otras. 2 casos mínimo para cada caso:** Todos los métodos *get* usan la referencia this, para desambiguar los atributos de instancias de otros parámetros.

MANUAL DE USUARIO:

Al abrir la aplicación el usuario es bienvenido en una ventana la cual le da su mensaje de bienvenida, como se pidió en el trabajo, al dar click en el nombre de cada uno se cambian las fotografías asociadas al integrante del grupo. Esta ventana tiene un botón llamado Entrar el cual sirve para acceder al menú principal

Dentro de esta ventana hay un menú el cual tiene Inicio, Procesos, Consultas, Eliminar y Ayuda, las cuales son las funcionalidades que requiere el CRM.

Archivo tiene una información sobre el programa, en la opción Archivo tendrá información de la aplicación y en la opción salir ejecutará un comando para terminar el proceso

En **Procesos** tenemos las opciones para crear empleados, cliente, empresas, servicio y negocio.

Para crear un cliente se deben llenar unos contenedores con la información correspondiente al nuevo cliente, entre los cuales se encuentran

Consultas tendrá todo lo relacionado con la información del sistema de clientes, empleados, empresas y negocios

Eliminar contendrá todo lo relacionado para eliminar clientes, empleados, empresas y negocios