****

**Universidad Latina de Costa Rica**

**Programación IV**

**Control de tiempos para actividades laborales**

**Modelo UML**

Marco Valverde Mejía

**Profesor**

Jonathan Castro Ramírez

José Zúñiga Martínez

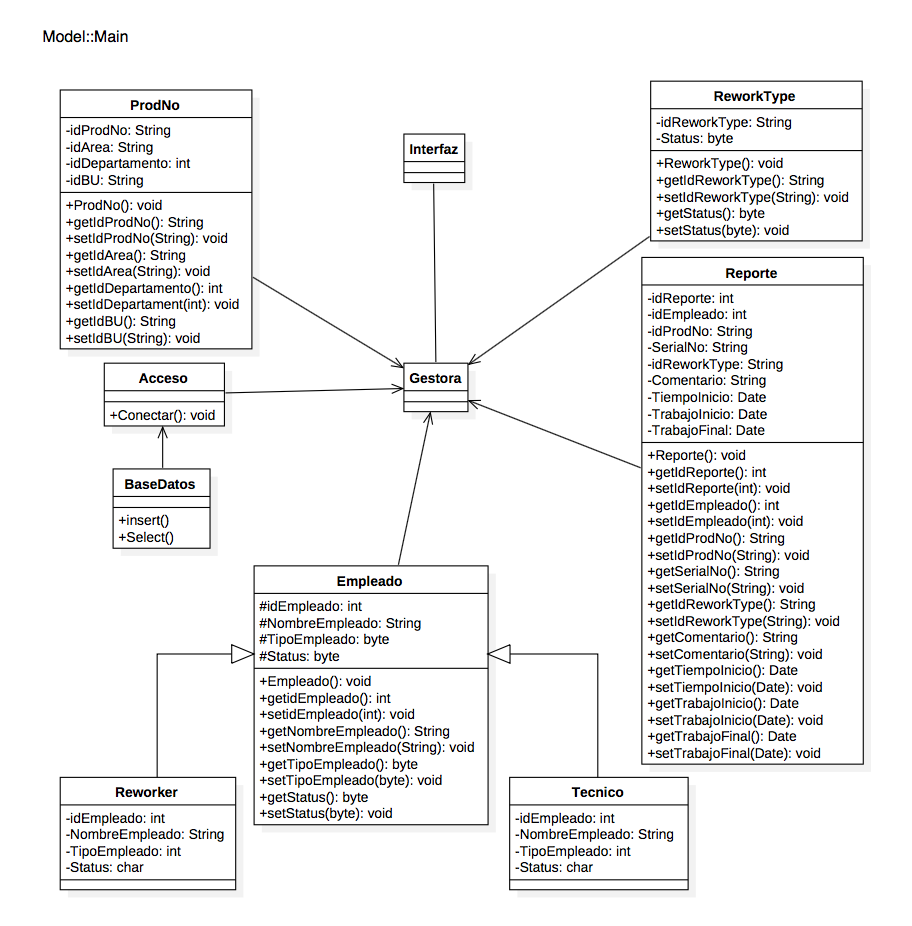
Fabián Rodríguez Meza

Alonso Mesen Quesada

**Estudiantes**

**2017**

**San Pedro**

**Documentación UML**

**¿Por qué el modelo de Programación Orientada a Objeto?**

En este proyecto, utilizamos el modelo de Programación Orientada a Objetos debido a que, por su facilidad de uso, nos permite realizar el programa de una forma más organizada y simple para trabajar en él, tanto para la lectura como escritura de datos, haciendo uso de métodos como la abstracción, interface, herencia y polimorfismo.

**Explicación modelo UML**

**Clase Interfaz:**

En esta parte se trabajará con todo la parte grafica del programa, esta clase seria la ventana que el usuario vería ocultándole todo el código, permitiéndole y asegurándonos de que trabajará sin problemas futuros por errores de modificación de código.

**Clase ProdNo:**

En esta clase, se leerá y escribirá la información acerca de los Números de Producto de la empresa, haciendo uso de métodos como Get() y Set() y el constructor para instanciarla.

**Clase Empleado:**

En esta clase, se obtendrán atributos que serán utilizados en dos clases mas, haciendo uso de la Herencia

**Clase Reworker:** Es un tipo de empleado el cual realizara la tarea indicada por el técnico, hace uso de la herencia

**Clase Técnico:** Es un tipo de empleado el cual le indica al Reworker, que tarea realizar para los productos de la empresa, hace uso de la herencia

**Clase ReworkType:**

Clase donde se escribirán y leerán las tareas que el técnico le indica al Reworker que debe realizar, se hace uso de los métodos Get() y Set().

**Clase Reporte:**

En esta clase, se obtienen todos los datos en forma de reporte de todas las tarjetas electrónicas procesadas, también hace uso de los métodos Get() y Set().

**Base de datos:**

Base de datos donde se crearan todas las tablas para hacer uso de la información guardada en ella por el programa, de aquí se puede leer y escribir en el programa.

**Clase Acceso:**

Esta clase, es la comunicación entre el programa y la base de datos, evitando que cada método tenga que hacer una conexión con la base de datos, haciendo que la conexión se haga una sola y única vez.

**Clase Gestora:**

Esta clase, es una de las mas importantes debido a que es la clase que hace todas las comunicaciones entre todas las clases y la interfaz grafica del usuario, maneja todas las tareas del programa.