**Programación orientada a objetos en procesos continuos**

Carlos Alidio Medina López

Cod: 20191383031

1. **RESUMEN**

Se desea implementar una estructura con varios requisitos para la creación del programa y enfocado en la programación orientada a objetos. Para ser implementada en proceso continuos en la industria. Cada parte de la estructura se reviso en cada clase y se implemento en este proyecto final de la temática.

1. **PROCESO CONTINUO**

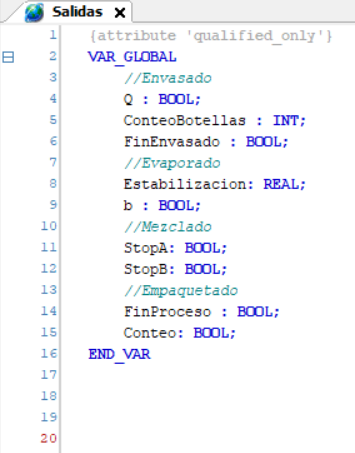
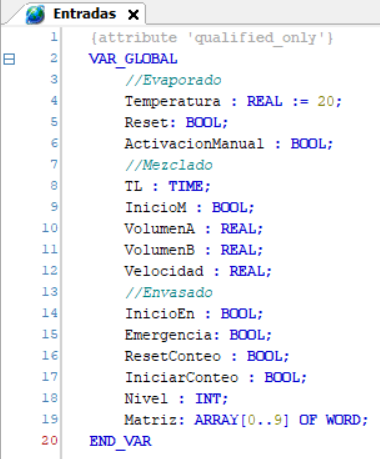
Para la creación del proyecto, se requiere de usar un proceso continuo, se encontró un proceso de destilación el cual se divide en tres etapas: Evaporar, mezclar y embotellamiento. Se puede observa el proceso completo en el siguiente diagrama P&ID.



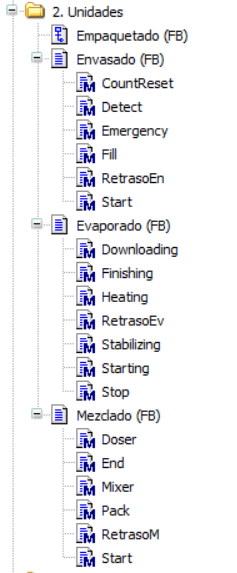
* 1. **Evaporar**: Cuenta con actuadores de bombas y válvulas, para permitir la circulación de la materia prima al tanque donde se va a realizar el proceso. Un intercambiador de calor, el que generara la temperatura necesaria para calentar la materia prima. Se requiere de una circulación de la materia, para la transferencia de temperatura correcta, para ello, se usa una bomba que permite la circulación del material, junto con un sensor de temperatura, para conocer la temperatura la cual posee la materia prima. Y finalmente cuando la materia prime llegue a la temperatura deseada, se usará una válvula y una bomba para circular el material desde el tanque de evaporación hasta la siguiente etapa.
  2. **Mezclar**: Recibiendo el material de la etapa anterior, se requiera agregar una nueva materia prima contenida en un tanque, el cual tiene una válvula y una bomba para la circulación y al tanque de mezcla. En el tanque de mezcla se encuentra un motor con un variador de velocidad, y se accionara cuando se halla depositado las dos materias primas. Finalmente, con una válvula y una bomba se descarga el material mezclado a la siguiente etapa.
  3. **Embotellamiento**: La etapa final que cuenta con el accionamiento de una banda transportadora para el desplazamiento de las botellas. Hay bomba y válvula para dosificar las botellas, y unos sensores de nivel.

1. **CODESYS**

Para la programación en Codesys, se crearon varios bloques para realizar la simulación y usar la programación orientada a objetos. Inicialmente se observa las variables globales en el proceso.



Se realizaron 4 unidades en la simulación, para la etapa de Evaporar, mezclar, envasar y empacar. Cada una con sus métodos.

**Evaporado**: Cuenta con 7 métodos:

* Starting: Inicio del proceso, activación de la válvula y bomba para llenar el tanque de evaporización con la materia prima.
* Heating: Activación de la bomba de circulación, intercambiador de calor y el extractor.
* Stabilizing: Inicia el conteo de evaporación.
* Downloading: Descarga de la materia prima a la siguiente etapa, activando la válvula y bomba.
* Finishing: Desactiva la válvula y bomba para detener la descarga de la materia prima.
* RetrasoEv: Genera un retraso para la simulación.
* Stop: Detiene la simulación.

**Mezclado**: Cuenta con 6 métodos:

* Start: Activa la válvula y la bomba de carga de la primera materia prima.
* Doser: Activa la válvula y la bomba de carga de la segunda materia prima.
* Mixer: Activación del motor para realizar la mezcla.
* Pack: Activación de la válvula y bomba de descarga.
* End: Finaliza el proceso, apagando la válvula y la bomba de descarga.

**Envasado**: Cuenta con 6 métodos:

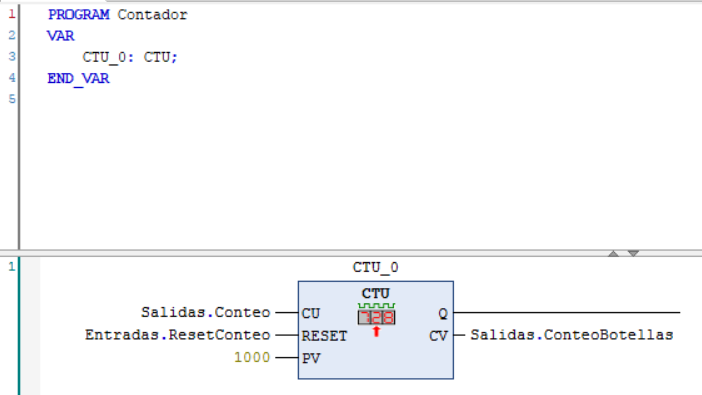
* Start: Activa la banda transportadora.
* Detect: Desactiva la banda transportadora y activa la válvula y la bomba que llena las botellas.
* Fill: Desactiva la válvula y la bomba, e inicia el contador.
* CountReset: Reinicio del contador de botellas.
* Emergency: Apagado de emergencia.
* RetrasoEn: Genera un retraso para la simulación.

**Empaquetado**: No cuenta con métodos, fue un bloque realizado en grafset.

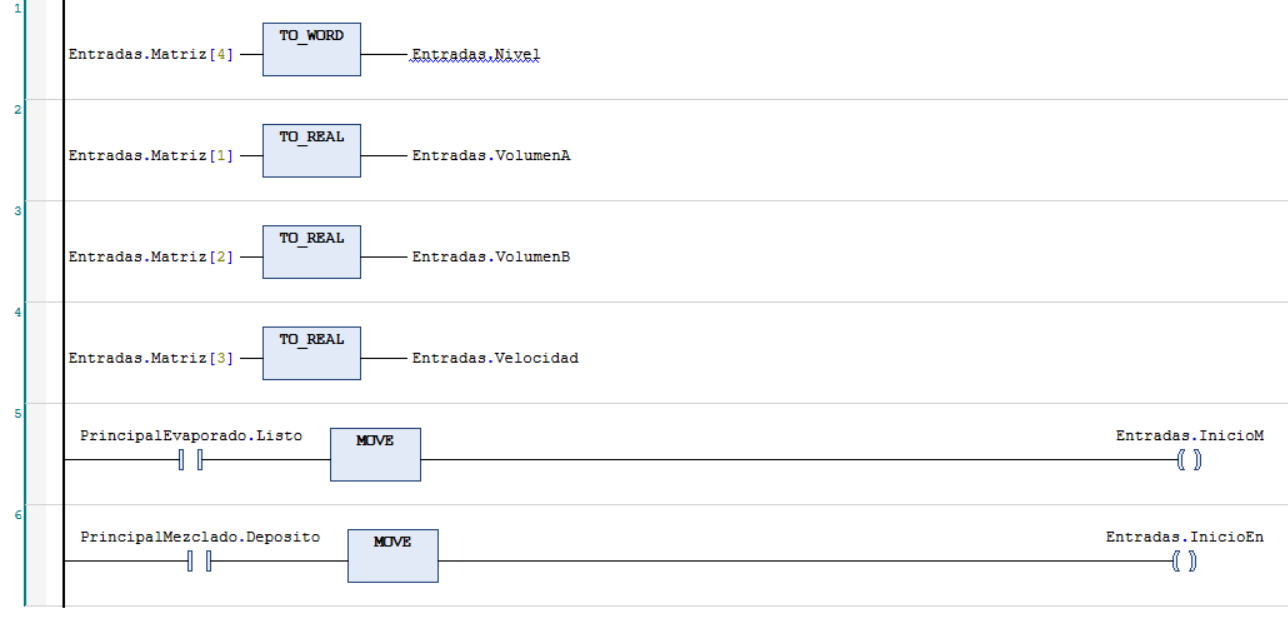
|  |
| --- |
| FUNCTION\_BLOCK Empaquetado  VAR\_INPUT  END\_VAR  VAR\_OUTPUT  END\_VAR  VAR  ProductoListo: BOOL;  DepositoVacio: BOOL;  Stop: BOOL;  DepositoLleno: BOOL;  SensorNivel: INT;  Valvula: BOOL;  Empujador: BOOL;  Listo: BOOL;  END\_VAR |
|  |

Se realizo algunos bloques de códigos principales, los que cumplen acciones principales en el programa.

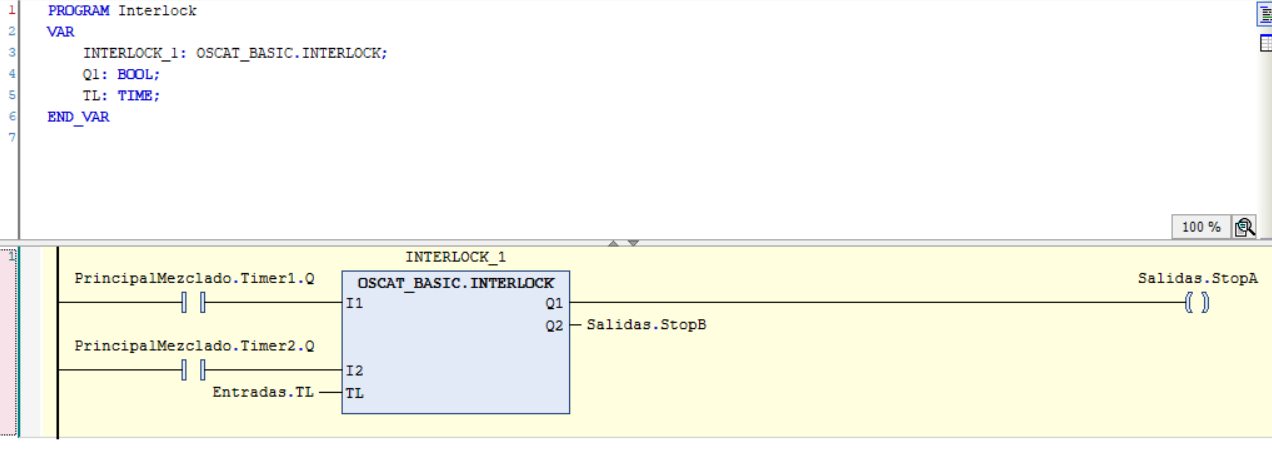
Un contador encargado en sumar la cantidad de botellas.



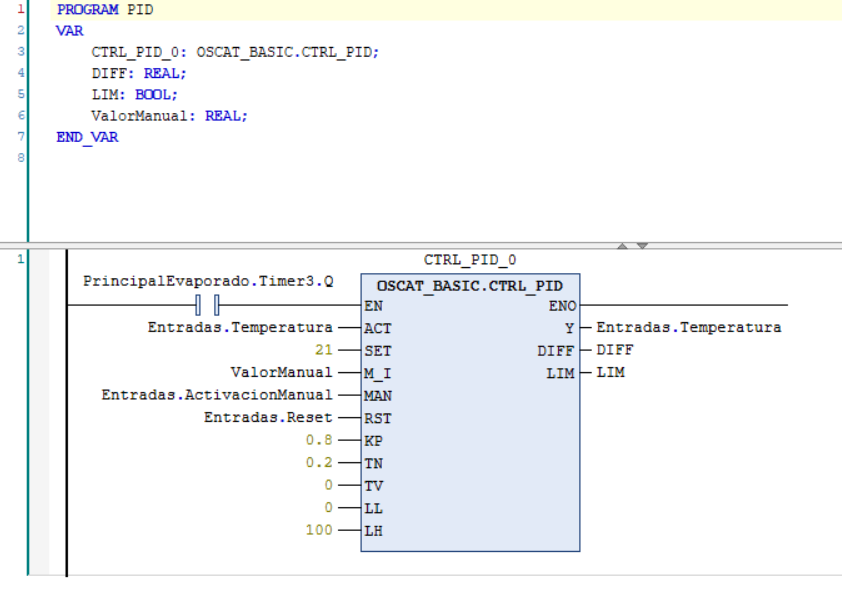
Transferencia de información de un vector a variables del proceso.



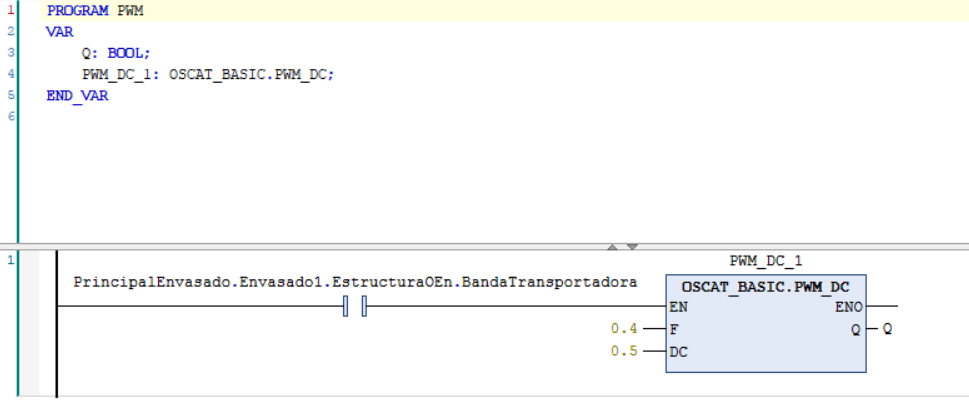
Un interlock de la librería OSCAT Basic



UN PID de la librería OSCAT Basic.



Un PWM de la librería OSCAT Basic



Bloque principal del proceso de evaporación usando los métodos anteriormente mencionadas.

|  |
| --- |
| PROGRAM PrincipalEvaporado  VAR  Wrapper0Ev : Wrapper1;  EntradasEv : EvaporadorI;  SalidasEv : EvaporadorO;  Inicio : BOOL;  MateriaPrima : BOOL;  EspacioTanque : BOOL;  Evaporate : Evaporado;  Tiempo : BOOL;  Estabilizacion : BOOL;  EspacioDeposito : BOOL;  Timer : TON;  Timer1 : TON;  Timer2 : TON;  Timer3 : TON;  Timer4 : TON;  Timer5 : TON;  Listo : BOOL;  Texto : STRING;  END\_VAR |
| Wrapper0Ev(EstructuraIEv:=EntradasEv, EstructuraOEv=>SalidasEv);  IF Salidas.FinProceso:=TRUE THEN  Inicio:=TRUE;  Salidas.FinProceso:=FALSE;  IF (Entradas.Matriz[0] >= 51) THEN  Texto:='SOBRECARGA';  Evaporate.Stop();  END\_IF  IF (Entradas.Matriz[0] <= 10) THEN  Texto:='TANQUE VACÍO';  Evaporate.Stop();  END\_IF  IF (Entradas.Matriz[0]>=11 AND Entradas.matriz[0]<=50) THEN  Texto:='Materia Prima disponible';  Listo:=FALSE;  Timer(IN:=Inicio, PT:=T#3S);  IF Timer.Q THEN  Timer.IN:=FALSE;  MateriaPrima:=TRUE;  Evaporate.Starting();  Timer1(IN:=TRUE, PT:=T#3S);  END\_IF  IF Timer1.Q THEN  Timer1.IN:=FALSE;  MateriaPrima:=FALSE;  EspacioTanque:=TRUE;  Evaporate.Heating();  Timer2(IN:=EspacioTanque, PT:=T#3S);  END\_IF  IF (Entradas.Temperatura >= 23) AND (Timer2.Q) THEN  Timer2.IN:=FALSE;  EspacioTanque:=FALSE;  Estabilizacion:=TRUE;  Evaporate.Stabilizing();  Timer3(IN:=Estabilizacion, PT:=T#5S);  END\_IF  IF Timer3.Q THEN  Timer3.IN:=FALSE;  Estabilizacion:=FALSE;  Tiempo:=TRUE;  Evaporate.Downloading();  Timer4(IN:=Tiempo, PT:=T#3S);  END\_IF  IF Timer4.Q THEN  Timer4.IN:=FALSE;  Tiempo:=FALSE;  EspacioDeposito:=TRUE;  Evaporate.Finishing();  Timer5(IN:=EspacioDeposito, PT:=T#5S);  END\_IF  IF Timer5.Q THEN  Timer5.IN:=FALSE;  EspacioDeposito:=FALSE;  Listo:=TRUE;  END\_IF  END\_IF  END\_IF |

Bloque principal del proceso de mezcla usando los métodos anteriormente mencionadas.

|  |
| --- |
| PROGRAM PrincipalMezclado  VAR  Wrapper0M : Wrapper2;  EntradasM : MezcladorI;  SalidasM : MezcladorO;  Mezclado1 : Mezclado;  EspacioTanque : BOOL;  Timer : TON;  Timer1 : TON;  Timer2 : TON;  Timer3 : TON;  Timer4 : TON;  Timer5 : TON;  Deposito : BOOL;  Proceso:BOOL;  Proceso1:BOOL;  Proceso2: BOOL;  Proceso3: BOOL;  Proceso4: BOOL;  END\_VAR |
| Wrapper0M(EstructuraIM:=EntradasM, EstructuraOM=>SalidasM);  IF Entradas.InicioM THEN  Proceso:=TRUE;  Timer(IN:=Proceso, PT:=T#2S);  Entradas.TL:=T#3S;  IF Timer.Q THEN  Timer.IN:=FALSE;  EspacioTanque:=TRUE;  Mezclado1.Start();  Timer1(IN:=EspacioTanque, PT:=T#3S);  END\_IF  IF Timer1.Q AND Entradas.VolumenA >= 2 THEN  Timer1.IN:=FALSE;  EspacioTanque:=FALSE;  Proceso1:=TRUE;  Mezclado1.Doser();  Timer2(IN:=Proceso1, PT:=T#3S);  END\_IF  IF Timer2.Q AND Entradas.VolumenB >= 1.5 THEN  Timer2.IN:=FALSE;  Proceso1:=FALSE;  Proceso2:=TRUE;  Mezclado1.Mixer();  Timer3(IN:=Proceso2, PT:=T#3S);  END\_IF  IF Timer3.Q AND Entradas.Velocidad >= 15 THEN  Timer3.IN:=FALSE;  Proceso2:=FALSE;  Proceso3:=TRUE;  Timer4(IN:=Proceso3, PT:=T#5S);  END\_IF  IF Timer4.Q THEN  Timer4.IN:=FALSE;  Proceso3:=FALSE;  Proceso4:=TRUE;  Mezclado1.Pack();  Timer5(IN:=Proceso4, PT:=T#2S);  END\_IF  IF Timer5.Q THEN  Timer5.IN:=FALSE;  Proceso4:=FALSE;  Deposito:=TRUE;  Mezclado1.End();  END\_IF  END\_IF |

Bloque principal del proceso de envasar usando los métodos anteriormente mencionadas.

|  |
| --- |
| PROGRAM PrincipalEnvasado  VAR  Wrapper0En : Wrapper3;  EntradasEn : EnvasadorI;  SalidasEn : EnvasadorO;  Envasado1 : Envasado;  Timer1: TON;  Timer2: TON;  Timer3: TON;  Timer4: TON;  Proceso1 : BOOL;  Proceso2: BOOL;  Proceso3 : BOOL;  Conteo : INT;  END\_VAR |
| Wrapper0En(EstructuraIEn:=EntradasEn, EstructuraOEn=>SalidasEn);  IF Entradas.InicioEn THEN  Proceso1:=TRUE;  Envasado1.Start();  Timer1(IN:=Proceso1, PT:=T#3S);  IF Timer1.Q AND Entradas.Nivel >= 5 THEN  Timer1.IN:=FALSE;  Proceso1:=FALSE;  Proceso3:=FALSE;  Proceso2:=TRUE;  Envasado1.Fill(Conteo);  Entradas.Matriz[4] := 0;  Timer2(IN:=Proceso2, PT:=T#3S);  Salidas.FinEnvasado:=TRUE;  END\_IF  IF Timer1.Q AND Entradas.Nivel < 5 THEN  Timer1.IN:=FALSE;  Proceso1:=FALSE;  Proceso2:=FALSE;  Proceso3:=TRUE;  Envasado1.Detect();  Timer3(IN:=Proceso3, PT:=T#3S);  END\_IF  IF Entradas.Emergencia THEN  Envasado1.Emergency();  END\_IF  IF Entradas.ResetConteo THEN  Envasado1.CountReset();  END\_IF  END\_IF |

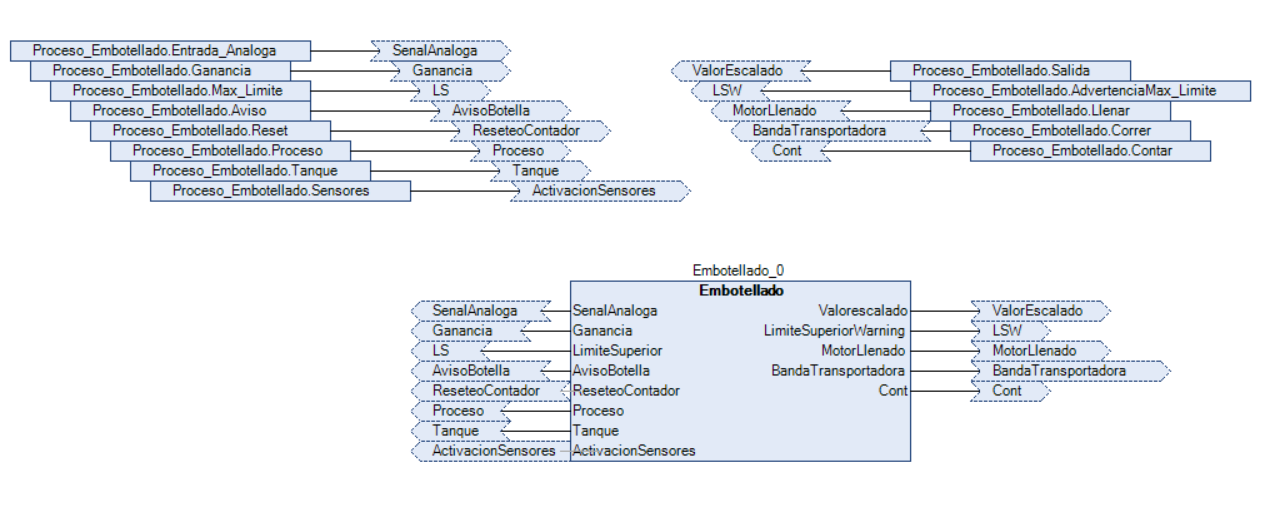
Bloque principal del proceso de empacar llamando el bloque de grafset.

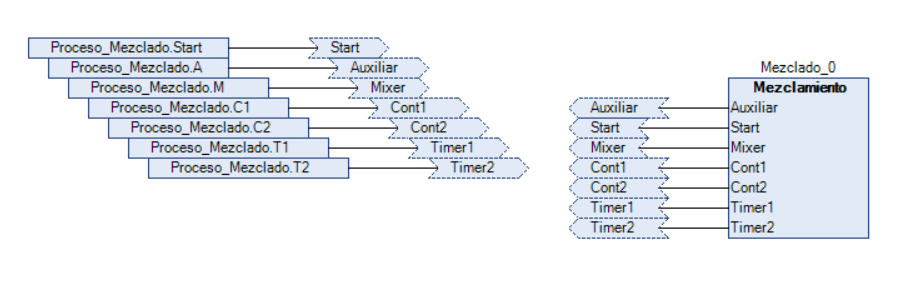
|  |
| --- |
| PROGRAM PrincipalEmpaquetado  VAR  Empaquetado\_0: Empaquetado;  END\_VAR |
|  |

En la sección de DUT se almacenaron las estructuras creadas para cada unidad con sus respectivas entradas y salidas.

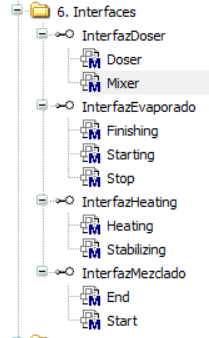
|  |  |
| --- | --- |
| TYPE EvaporadorI :  STRUCT  Temperatura: REAL;  END\_STRUCT  END\_TYPE | TYPE EvaporadorO :  STRUCT  BombaCarga : BOOL;  ValvulaCarga : BOOL;  Calefactor : BOOL;  BombaRecirculacion : BOOL;  Extractor : BOOL;  ValvulaDescarga : BOOL;  BombaDescarga : BOOL;  End1 : BOOL;  EmpezarTiempoEvaporacion : BOOL;  b : BOOL;  END\_STRUCT  END\_TYPE |
| TYPE MezcladorI :  STRUCT  Evaporador : Evaporado;  END\_STRUCT  END\_TYPE | TYPE MezcladorO :  STRUCT  ValvulaCargaA: BOOL;  BombaCargaA: BOOL;  ValvulaCargaB: BOOL;  BombaCargaB: BOOL;  Mezclar: BOOL;  ValvulaDescarga1 : BOOL;  BombaDescarga1 : BOOL;  Stop: BOOL;  END\_STRUCT  END\_TYPE |
| TYPE EnvasadorI :  STRUCT  F : REAL;  DC : REAL;  END\_STRUCT  END\_TYPE | TYPE EnvasadorO :  STRUCT  BandaTransportadora : BOOL;  ValvulaLlenado : BOOL;  BombaLlenado : BOOL;  Q : BOOL;  END\_STRUCT  END\_TYPE |
| TYPE DatosEstructura :  STRUCT  Matriz: ARRAY[0..9] OF WORD;  END\_STRUCT  END\_TYPE | |

Se creo un componente, siendo el programa inicia y donde se realiza la conexión de variables de mezcla y envasado.





Se implemento interfaces para las unidades con los métodos principales.



Y por ultimo se presenta la creación de lo Wrapper, que se utilizaron anteriormente en los bloques principales.

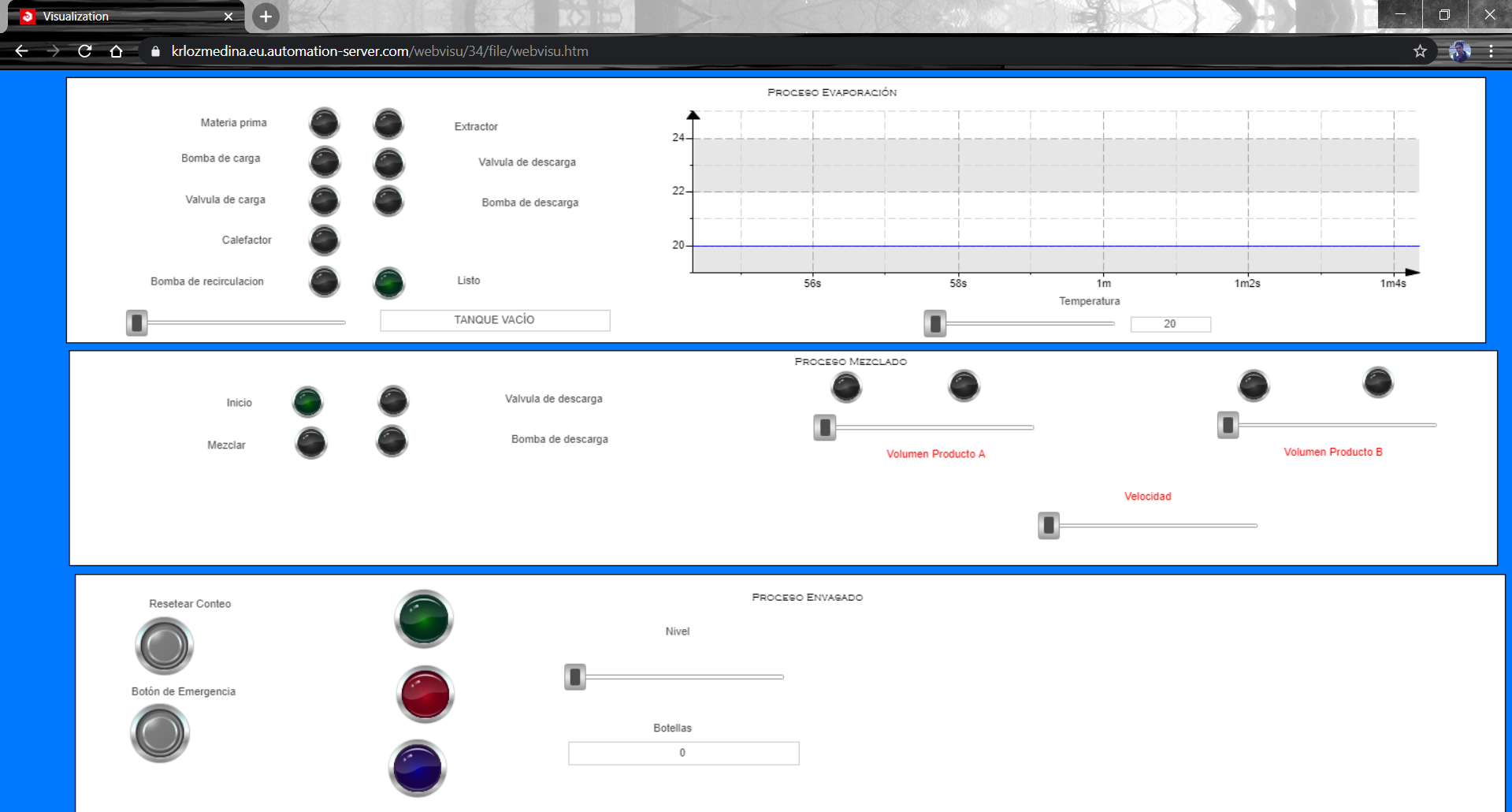
|  |
| --- |
| FUNCTION\_BLOCK Wrapper1  VAR\_INPUT  EstructuraIEv : EvaporadorI;  END\_VAR  VAR\_OUTPUT  EstructuraOEv : EvaporadorO;  END\_VAR  VAR  ConveyorEv : Evaporado;  END\_VAR |
| ConveyorEv(EstructuraIEv:=EstructuraIEv, EstructuraOEv=>EstructuraOEv);  ConveyorEv.RetrasoEv(RetrasoUpEv:=T#5S);  ConveyorEv.Starting(); |

|  |
| --- |
| FUNCTION\_BLOCK Wrapper2  VAR\_INPUT  EstructuraIM : MezcladorI;  END\_VAR  VAR\_OUTPUT  EstructuraOM : MezcladorO;  END\_VAR  VAR  ConveyorM : Mezclado;  END\_VAR |
| ConveyorM(EstructuraIM:=EstructuraIM, EstructuraOM=>EstructuraOM);  ConveyorM.RetrasoM(RetrasoUpM:=T#5S);  ConveyorM.Start(); |

|  |
| --- |
| FUNCTION\_BLOCK Wrapper3  VAR\_INPUT  EstructuraIEn : EnvasadorI;  END\_VAR  VAR\_OUTPUT  EstructuraOEn : EnvasadorO;  END\_VAR  VAR  ConveyorEn : Envasado;  END\_VAR |
| ConveyorEn(EstructuraIEn:=EstructuraIEn, EstructuraOEn=>EstructuraOEn);  ConveyorEn.RetrasoEn(RetrasoUpEn:=T#5S);  ConveyorEn.Start(); |

1. **HMI Y AUTOMATION SERVER**

Se creo una visualización simple, donde por medio de bombillos se simula la activación de válvulas y bombas, además de algunas sliders para ingresar valores tal como la temperatura y niveles de los tanques.



1. **CONCLUSIONES**

Aunque parezca tedioso la implementación de la estructura, durante la realización del proyecto se nota la importancia de cada modulo ya que genera inicialmente un orden y encontrar fácilmente errores en el código, además de realizar pruebas independientes de los módulos. Aunque el proyecto fallo al no implementarse sobre los actuadores y sensores, y desviarse un poco en la programación orientada a objetos, aun así se logro implementar los módulos vistos en clase.