

# Clases

## Clase Logger

Es la clase que logea todas las acciones del sistema. Es singleton, ya que solo va a haber un log, y es a la vez un visitor, ya que encapsula la funcionalidad del logear utilizando polimorfismo en cada clase que lo llama. Dentro del “Accept” de cada método hay un “logger.loggear(this)”. Dentro del logger, hay varios métodos llamados “loggear” con sobrecarga, cada uno con el parámetro de la clase sobre la que desea actuar.

## Clase Estrategia Asignación, PrimeroPreferidos, SectorPrioritario y TodosAsignados

Es una clase abstracta que engloba a las clases PrimeroPreferidos, SectorPrioritario y TodosAsignados.

Son clases diferentes porque cada una tiene una funcionalidad específica. Por ejemplo, la clase “SectorPrioritario” tiene el método “sectoresPrioritarios” que devuelve una lista con los sectores prioritarios. También, el método “asignarHorarios()” está overrideado en cada subclase.

## Clase Empleado

Es una clase abstracta que engloba a las clases Operario y Supervisor. Son diferentes clases ya que una difiere de la otra por el atributo “tienePermisoDeConfirmacion”.

## Clase Supervisor

Tiene el atributo booleano “tienePermisoDeConfirmacion” que, dependiendo de si el supervisor puede o no confirmar la asignación de horario, retorna True o False. Decidimos no hacer dos clases “Supervisor” ya que era agregar complejidad sin obtener ningún beneficio a cambio.

## Clase Estrategia Notificación, SendEmail, SendSMS, y SendWP

La clase “Estrategia Notificación” es abstracta y engloba a las clases SendEmail, SendSMS y SendWP.

Éstas clases overridean el método “execute()” que heredan de Estrategia Notificación, ya que según la consigna, cada uno de ellos tiene diferentes funcionalidades.

# Métodos

## Método asignarHorarios()

Este método le asigna a los empleados un horario de acuerdo al criterio especificado en cada subclase de “Estrategia Asignacion”.

## Método notificar()

Este método notifica la asignación horaria a diversas personas. Está dentro de la clase sector porque cada sector puede definir sus propios medios de notificación. Es un wrapper que llama al “execute()” de los elementos dentro de la lista “Estrategias Notificación”.

## Método confirmarHorario(Supervisor)

Este método se fija que el supervisor que quiere confirmar los horarios tenga el permiso de hacerlo. Lo hace a través del atributo “tienePermisoDeConfirmacion” del supervisor, y si es True, lo permite.