

Calendario(Contenedor)
Reservacion* reservacion = nullptr Nodo* nodoSiguiente = nullptr  Nodo(Reservacion*, Nodo*) ~Nodo()  Reservacion* getReservacion() const void setReservacion(Reservacion*) Nodo* getNodoSiguiente() const void setNodoSiguiente(Nodo*)
Nodo* primeraReservacion = nullptr time_t segundosActual struct tm fechaActual int cantidadReservaciones = 0 Calendario() ~Calendario() time_t getSegundosActual() const struct tm getFechaActual() const int getCantidadReservaciones() const Nodo* getNodoPrimeraReservacion() bool nuevaReservacionLogica(struct tm, struct tm) const bool choqueConCalendario(struct tm, struct tm) const int preguntarAnno() const int preguntarMes() const int preguntarDia() const int preguntarHora() const int preguntarMinuto() const  struct tm preguntarFechaGeneral() const struct tm preguntarFechaEspecifica(string) const void imprimirCalendarioCompleto()

ListaUsuario(contenedor)
Nodo* next = nullptr Usuario* usuario Nodo(Usuario*, Nodo*) ~Nodo() Nodo* getNext() const void setNext(Nodo*) Usuario* getUsuario() const void setUsuario(Usuario*)  Nodo* head = nullptr
ListaUsuario() ~ListaUsuario() void invitar(Reservacion*) bool vacia() bool comprobarID(int) bool IdEsContribuidor(int) bool hayContribuidores() void eliminar(time_t); void agregarUsuario(Usuario*) Usuario* UsuarioPorID(int) void eliminarContribuidorPorID(int) void imprimir(string, string) void imprimirIDS() Usuario* verificador(const std::string&, const std::string&) bool verificador2(const int&, const std::string&)  void escribirEnArchivo(std::ofstream&, ofstream&) void escribirIdsEnArchivo(std::ofstream&) void leerIds(std::ifstream&, ListaUsuario*)

### Interaccion(Manejo del Programa)

ListaUsuario\* usuariosRegistrados  
Usuario\* usuarioActivo = nullptr  
Manager\* manActivo = nullptr

Interaccion()  
~Interaccion()

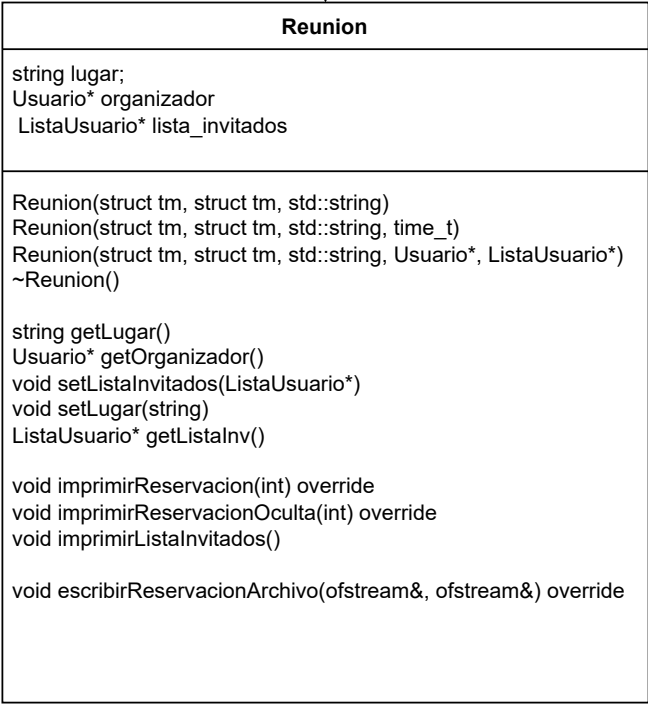
int menu\_entrada()  
string pedirTexto(string)  
int pedirCedula()  
int pedirPuesto()  
int menuAccionContribuidor()  
int menuAccionManager()

Usuario\* crear\_Usuario()  
bool iniciar\_sesion()  
void realizarAccionContribuidor()  
void realizarAccionManager()

void escribirEnArchivo()  
bool guardar\_en\_archivo\_cal(int)  
void leerArchivo(std::ifstream&)  
void leerManager(std::ifstream&)  
void leerContribuidor(std::ifstream&)

std::string codificar(std::string)

void ejecutar()



```
void imprimirCalendarioCompleto()
void imprimirCalendarioOculto()
int menuReservaciones() const
void crearReservacion(Usuario*, ListaUsuario*)
ListaUsuario* printParaAgregarId(Usuario*, ListaUsuario*)
void ordenarReservacion(Reservacion*)
Reservacion* obtenerReservacionPorPosicion(int);Add commentMore
actions
    string posicionEs(int);
    void eliminarReservacionIndividual(int);
    void eliminarReservacionSocial(int, int);
    void eliminarReservacion(time_t);

void escribirEnArchivo(ofstream&, ofstream&) const
struct tm* leerIdYFechasDeArchivo(ifstream&)
Reservacion* leerReunionDeArchivo(ifstream&)
Reservacion* leerActividadSocialDeArchivo(ifstream&)
Reservacion* leerEventoDiarioDeArchivo(ifstream&)
```

Main
Corre el codigo desde ejecutar de interaccion

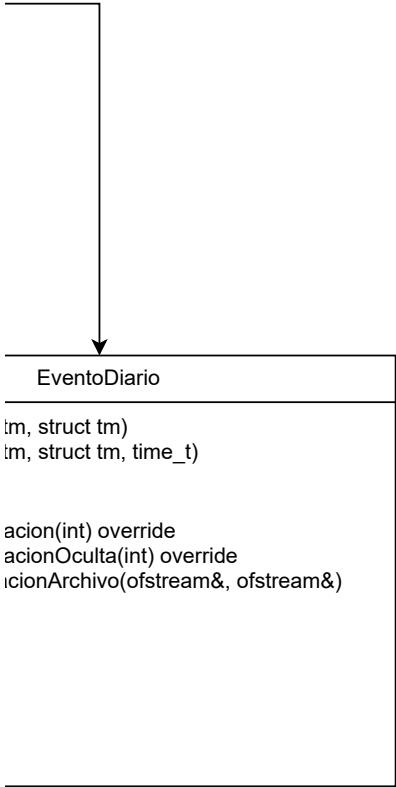
Comprobacion(Manejo de errores)
bool cadena_vacia(std::string) bool cadena_con_punto(std::string) bool cadena_con_espacio(std::string) bool de_string_a_int(std::string) bool numero_fuera_de_rango(int, int, int)  bool texto_no_vacio_sin_espacios(std::string) bool numero_entero_dentro_de_rango(int, int, std::string) bool numero_entero_sin_rango(std::string)  bool annoValido(int) bool mesValido(int) bool diaValido(int) bool horaValida(int) bool minutoValido(int)

Reservacion (Abstracta)
struct tm fechalnicio struct tm fechaFin time_t idReservacion
Reservacion(struct tm, struct tm) Reservacion(struct tm, struct tm, time_t) ~Reservacion()  struct tm& getFechalnicio() struct tm& getFechaFin() void setFechalnicio(struct tm) void setFechaFin(struct tm)  virtual void imprimirReservacion(int) = 0 virtual void imprimirReservacionOculto(int) = 0 string getFecha(struct tm&)  void escribirIdYFechas(ofstream&) virtual void escribirReservacionArchivo(ofstream&, ofstream&) = 0  bool choqueFechas(struct tm&, struct tm&)

ActividadSocial
sta_org ir
/(struct tm, struct tm, time_t) /(struct tm, struct tm, ListaUsuario*) al()  servacion(int) override servacionOculto(int) override servacionArchivo(std::ofstream&, ofstream&)  iraOrganizadores()

CitaPersonal
CitaPersonal(struct tm, struct tm) CitaPersonal(struct tm, struct tm, time_t) ~CitaPersonal()  void imprimirReservacion(int) override void imprimirReservacionOculto(int) override void escribirReservacionArchivo(ofstream&, ofstream&) override

EventoDiario(struct t EventoDiario(struct t ~EventoDiario()  void imprimirReserv void imprimirReserv void escribirReserva override
---



Reservacion
time_t idReservacion
getIdReservacion()

Reunion
Usuario* organizador ListaUsuario* invitados

ActividadSocial
ListaUsuario* organizadores

Calendario
+ field: type
crearReservacion(){ Reunion: -Pedir ids invitados -Agregarlo a invitados }