26/11/18

* Se creo el proyecto y se crearon y establecieron los logos
* Comienzo de adaptación para soporte multidioma

27/11/18

* Diseño de la interfaz de la pestaña Configuracion

28/11/18

* Funcionamiento correcto de la interfaz de Configuracion salvo petición de confirmación al servidor
* Busqueda en internet prueba e implementación de las clases Toast y Spinnerviewcontroller para generar avisos y esperar por el servidor

29/11/18

* Creacion de algoritmos para encriptar y desencriptar la clave cuando va por la red
* Modificacion en los php para desencriptar desde la app que venga encriptarlo en los propios script para escribir en el fichero (segunda capa de cifrado)
* Creación de nuevo php para coneccion desde mac
* Imprementacion de conexión con la app
* Implementacion de una LoadingView con una biblioteca y sustituyendo asi al SpinnerViewController (SUPUTAMADRE) e instalación de cocoapods
* A falta de implementar las SharedReferences para acabar esta pestaña

30/11/18

* Programación de la clase MyUserDefauld
* Finalización de la pestaña configuración

01/12/18

* Se han añadido las tab dse todas las opciones
* Se han programado las de group
* Se ha creado la clase Generic para objetos genericos
* Se han añadido a las traducciones errores
* Todas las tab funcionan pero por debajo es una chapuza gorda
* Arreglada la chapuza con un delegate en todas las pestañas se intentara reducir el código de la pantalla de espera unificándolo con la clase toast
* Arreglado todo se dan por concluida la pantalla de inicio y sus respectivas pestañas, se pasara a empezar con el grupo de trabajo

02/12/18

* Diseño de la interfaz de grupo de trabajo
* En la pantalla del Groupviewcontroller se ha colocado una navigationview y un botón add parta poder hacer el segue que tambiens e han creado
* Diseño definitivo de dicha pestaña

03/12/18

* Rediseño de las pestañas principales añadiento en todas el navigation bar y el botton add

08/12/18

* A GroupViewController se le ha añadido la posibilidad de refrescar la tabla
* Se ha creado la clase modelo Worker, que lleva la información de los trabajadores del proyecto
* Se ha creado el método con get worker y got worker para obtener de la base de datos trabjadores
* Se han configurado en programación los segues de forma que funcionen de manera optima
* Se ha configurado la clase DataGroupController de forma que solo es necesario configurar el botón guardar y la pregunta del Done

09/12/18

* Creada función para convertir de forma segura las URL
* Se ha añadido el botón que nos permite borrar trabajadores
* Se ha implementado las funciones de añadir o modificar si es necesario y la de eliminar
* Toast de txt vacion en conffragment
* Creación de variables globales y finales de estados para claridad de código
* Adaptacion del código para esto
* Creación de flag y de willdidapear para q al volver del segue se llamen de nuevo a los cambios que haya podido haber

10/12/18

* Tabla con source siendo nsmutablearray y todo lo que conlleva (origen búsqueda etc) y adaptación de esto en todas las tablas
* Se ha completado la programación de los paquetes y se ha creado en la clase utils una función llamada showError static y publica para evitar repetir código con la consecuente adaptación del proyecto a la utilización de esa función
* Se ha comenzado a configurar la implementación de Actores
* Se ha diseñado la pestaña datos de actores
* Creacion de una sola clase para todos los protocols
* Creacion de clase Actor y getActor para obtener actor de base de datos
* Creacion del resto de clases modelos y protocolos
* Creacion del resto de gets para obtener objetos de la base de datos, falta codificarlos

11/12/18 (Empece a las 15:30 y el dia 12 acabe de trogramas a las 20 casi tod seguido)

* Implementacion de las funciones get para obtener los objetos e implementación de descripption en los objetos para poder depurar bien
* Rellenada la pestaña de datos de actor
* Rellenadas las pestañas de autores y fuentes
* Unwind para llevar datos hacia atrás
* Diversas pruebas sobre tablas marcados y de mas cosas
* Creacion en generic de una variables para seleccionar
* Me he dado cuenta que es posible con una clase hacer todo y reducir aun mas el código
* Se procede a adaptación del proyecto (Puede durar muchop tiempo)
* **No se puede la adaptación porq si no entran todas las pestañas en la tab se asignan mal los valores asiq se reducirá por otro lado**
* **Se me ha ocurrido que empezando los objetivos intentar reutilizar las clases AutorActorCollection y Sources… Cambiando nombre porq será necesario y la de AddItems Actor Controller**

12/12/18

* Ya muestro los ítems para poder añadirlos
* ya decido cuando ejecutar segue o no
* ya añado a los actores fuentes y autores
* Actores temninado creo modifico destruyo
* Elimino autores y fuentes de actores
* Buena cominicacion entre todo ese confunto de pestañas
* Geturlsave movida hacia útil
* Geturldelete creada en útil
* Nuevo protocolo para obtener id de objetos cuando se guardan para guardar sus autores fuentes objetivos …..
* Se ha empezado la implementación de los **objetivos**
* **Solo se puede optimizar la clase additemcollector porq los autores fuentes y de mas como tienen vreada la variable tabbar para acceder a las variables de la tabbar padre no se puede reutilizar esas clases**
* YA recibo el objetivo y lo coloco en la interfaz (que ya he diseñado)
* También se ha diseñado el esqma de ventanas de esta parte
* Se han subido en el data los delegates y en los modelos se ha establecido valores predeterminados para prioridad etc y al subir los delegados cuando el diseño esta vacio empiezas en el valor predeterminado y no en el primero
* Objetivos completamente acabados mod cree li (ítems incluidos)
* Se ha comenzado a crear toda la interfaz de pantallas de recorrido para implementar los req no func
* Se ha terminado de diseñar toda la interfaz para los requisitos no funcionales

13/12/18

* Diseñada la interfaz de req no func
* Diseñada la interfaz de req fun

14/12/18

* Trabajando en la generalización de autores fuentes etc
* Autores y fuentes adaptado a una sola clase
* Objetivos en una sola clase
* Clase requisitos completada
* Requesitos no funcionales acabado
* Todas las clases reutilizables (autores, fuentes, obj y req) adaptadas para su uso
* Debido a un error no contemplado se ha metido la stackview de reqinfodata en dentro de una scroll view con todos los cambios que eso conlleva y el tiempo
* Todas las pantallas de datos pasadas a scrool por el tema de la visualización
* Comenzada la implementación de req de info (ya se reciben y se visualizan), se pueden añadir fuentes autores …. Solo falta guardar los datos específicos y el req info

15/12/18

* Creacion para guardar datos específicos sec normales …
* Adaptacion de un nuevo protocolo para guardar y eliminar ítems y asi no tener 2 protocolos (Se ha aprovechado para hacer una revisión y optimización del código
* Se ha creado la clase actorcontroler para poder añadir actores
* Implementada clase ReqFunTabBarController
* Implementadas las clases secnor y secexc controler
* Asignadas clases a las pestañas de reqfun
* Todos los elementos de interfaz unidos a todas las clases excepto a la de los datos
* Req funcionales funcioando y correctamente implementados

16/12/18

* revisión y ordanacion del codifo
* Traducion de la interfaz