

# Campaign-Sender & Campaign-File-Attacher

Este módulo se conforma por dos proyectos de bibliotecas de clases(**CampaignSender** y **CampaignFileAttacher**), un proyecto de Windows Form(**UIForm**) y otro de Pruebas Unitarias(**CampaignAndAttachTesting**).

Explicaré a detalle que realiza cada clase, así lograremos entrar a detalle que hace cada una.

## 1. CampaignSender

Cuenta con un total de 5 clases:

- **BaseCampaign:** Contendrá un atributo que será una lista de campañas así puedo tener encapsulado en una única clase las campañas y contendrá todos los métodos necesarios para obtener campañas del listado, obtener campañas a partir de un rango de tiempo de envío, sumar campañas y las validaciones pertinentes, esta clase terminó siendo reemplazada por la utilización de base de datos.
- **Campaign:** Va a contener todos los atributos(id, subject, body, listado de destinatarios y la fecha de envío) y lógica que forma parte de la clase como: Sumar un nuevo destinatario, sumar un nuevo listado de destinatarios, mostrar el listado y remover el listado de destinatarios así como la sobreescritura del método ToString.
- **Connection:** Se encarga de realizar la conexión con la base de datos. Realiza las consultas de "SELECT, INSERT y UPDATE", por cuestiones de tiempo no se siguió desarrollando más, pero almacena las campañas junto a la ruta con el adjunto almacenado.
- **Email:** Todos los destinatarios son instancias de la clase Email, que contiene dos atributos, sus propiedades y una sobreescritura.
- **SendMail:** Es la clase que se encarga de realizar el envío del email, realiza toda la lógica de conexiones con el Host y todos los datos necesarios para que se realice el envío de la campaña.

Hago una aclaración importante y es que, aunque demore en llegar, el mail si llega, si aparece el modal indicando "Envío realizado correctamente", por lo que hay que tener paciencia para ver el resultado ;)

## 2. CampaignFileAttacher

- **FileAttacher:** Cuenta con sus atributos y propiedades para poder crear instancias de esta clase, así como validar el tamaño de los adjuntos que en este caso no podrán ser mayor a 5mb.

## 3. UIForm

Cuenta con seis formularios, el principal, el formulario donde se muestra el listado de las campañas enviadas, menú de adjuntos, listado de campañas con adjuntos, cambiar o modificar un adjunto y menú de subida de adjunto:

- **AttachFileForm:** Contendrá el sub menú que conlleva tres opciones, subir archivo, listar campañas con adjuntos y cambiar o borrar un archivo.

**Campaign-File-Attacher**

Puedes subir archivos PDF a las campañas creadas, cambiar el archivo, borrar o únicamente saber que campañas tienen subido algún archivo adjunto

Subir Archivo...

Listado de Campañas c/ Adjuntos

Cambiar o Borrar Archivo

Salir

- **AttachmentListForm:** Comparte la misma estética que el listado de campañas, con la salvedad de que este trae únicamente el listado de campañas con adjuntos.

Desde: jueves , 01 julio , 20 Hasta: jueves , 01 julio , 20

Buscar

Salir

- **ChangeDeleteForm:** Hay que buscar la campaña que cuente con algún adjunto, cuando lo logré conseguir se tendrá la opción de poder cambiar el adjunto o eliminarlo directamente, donde también se modifica en la base de datos

Sube tu Archivo PDF a tu campaña seleccionada

Id de la campaña: 1 Buscar Campaña

Nombre:

Cambiar adjunto... Eliminar Adjunto

Salir

- **UploadForm:** Sube un archivo PDF únicamente con un peso máximo de 5mb a una campaña ya creada

- **MainFrm:** Los contactos a enviar el mail se debe escribir de a uno (no es lo más dinámico, lo sé jaja cuestiones de tiempo...), se debe indicar el asunto, el contenido y la fecha. El formulario cuenta con un Thread el cual va a informar el status de envío del email, y realmente explica bien lo que hace cada botón, este sería el formulario:

- **ScheduleForm:** Se encarga de mostrar todas las campañas que se mandaron en un lapso de tiempo dado, ejecutando los métodos de la base de datos, y trae los datos del Id, Subject y fecha.

#### 4. UnitTestingCampaignSender

Cuenta con cuatro pruebas unitarias simples, se tomaron en consideración y se escogieron con el propósito que varían en sí para corroborar que estuvieran correctas e incluso validar una excepción.

Prueba	Duración	F
✔ UnitTestingCampaignSender (..	3.8 s	
✔ UnitTestingCampaignSender	3.8 s	
✔ SenderTesting (4)	3.8 s	
✔ AddCampaignToBaseC...	34 ms	
✔ CompareIsTrueBecauseI...	< 1 ms	
✔ DoNotAddCampaignCa...	16 ms	
✔ SendACampaign	3.7 s	

#### FAQ

- No se logró funcionalidad al programar el envío ya que se le invirtió más dedicación a completar todo lo que pudiera lograr del challenge.
- Con más tiempo probablemente se le hubiese podido invertir más tiempo en la etapa de diseño para lograr conseguir implementar interfaces, eventos, herencia. Tengo presente que se puede implementar mejor, a pesar de contar con la funcionalidad.
- Se cuenta con una carpeta en la ruta **UIForm/bin/Debug/netcoreapp3.1/** llamada **FileAttached** donde se guardan los adjuntos que han sido subidos.
- A la par de donde se encuentra la solución se encuentra un archivo con el mismo nombre "PDC\_AlbertoGiffard" que sería del tipo .bak que es donde se encuentra la base de datos que se utilizó, habría que cambiar dentro de la clase "Connection" el nombre del server para que logre funcionar de forma correcta (Línea 19, el "." del Data Source).

Sería todo por el momento, gracias por leer ;)