Interfaz de usuario gráfica

Descripción generada automáticamente con confianza baja

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**FASE II**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

Índex

[1. INTRODUCCIÓ 2](#_Toc91671648)

[2. DESCRIPCIÓ DE LA PRÀCTICA 2](#_Toc91671649)

[3. INTRODUCCIÓ A APP DESIGNER 3](#_Toc91671650)

[4. FASE II 8](#_Toc91671651)

[4.1 GUI 10](#_Toc91671652)

[4.2 Processament a temps real 12](#_Toc91671653)

[5. ENTREGA 14](#_Toc91671654)

# 1. INTRODUCCIÓ

L'estudi de gravació de La Salle Campus Barcelona vol crear un plugin nou per l'eina d'edició sonora LSAudioTools que permeti equalitzar, reverberar o aplicar delay a les produccions realitzades pels alumnes i es pugui utilitzar en temps real en les entrevistes del canal de televisió SalleVisión. Per aquest motiu, s’han posat en contacte amb els estudiants d’enginyeria que cursen Processat de la Senyal perquè els ajudin a aconseguir-ho fent el primer prototip en Matlab.



**Per la realització de la Fase 2 de la pràctica és imprescindible que verifiqueu que disposeu de les llibreries següents: Signal Processing Toolbox, Audio Toolbox, DSP System Toolbox.**

# 2. DESCRIPCIÓ DE LA PRÀCTICA

La pràctica es dividirà en dues fases que s’han d’implementar de manera seqüencial pel correcte funcionament d’aquesta. Aquesta pràctica s’ha de realitzar en grup de 3 persones de la mateixa classe.

* **FASE I: Disseny dels diferents efectes (filtres IIR, reverberació i delay)**.
* **FASE II: Implementació d’una interfície d’usuari que funcioni a temps-real**

A l’hora de qualificar la pràctica es valorarà cadascuna de les fases per separat i la pràctica en conjunt tenint en compte que cada part es pondera amb un 50% de la pràctica. Es molt important complir amb els requeriments de cada fase i els requeriments finals de la pràctica.

|  |  |
| --- | --- |
| **Fase I** | 50% |
| **Fase II** | 50% |

L’entrega de la **primera fase** serà el dia **30/12/21 sobre 10** o bé 07/01/22 sobre 7 o bé **14/1/22 sobre 5.**

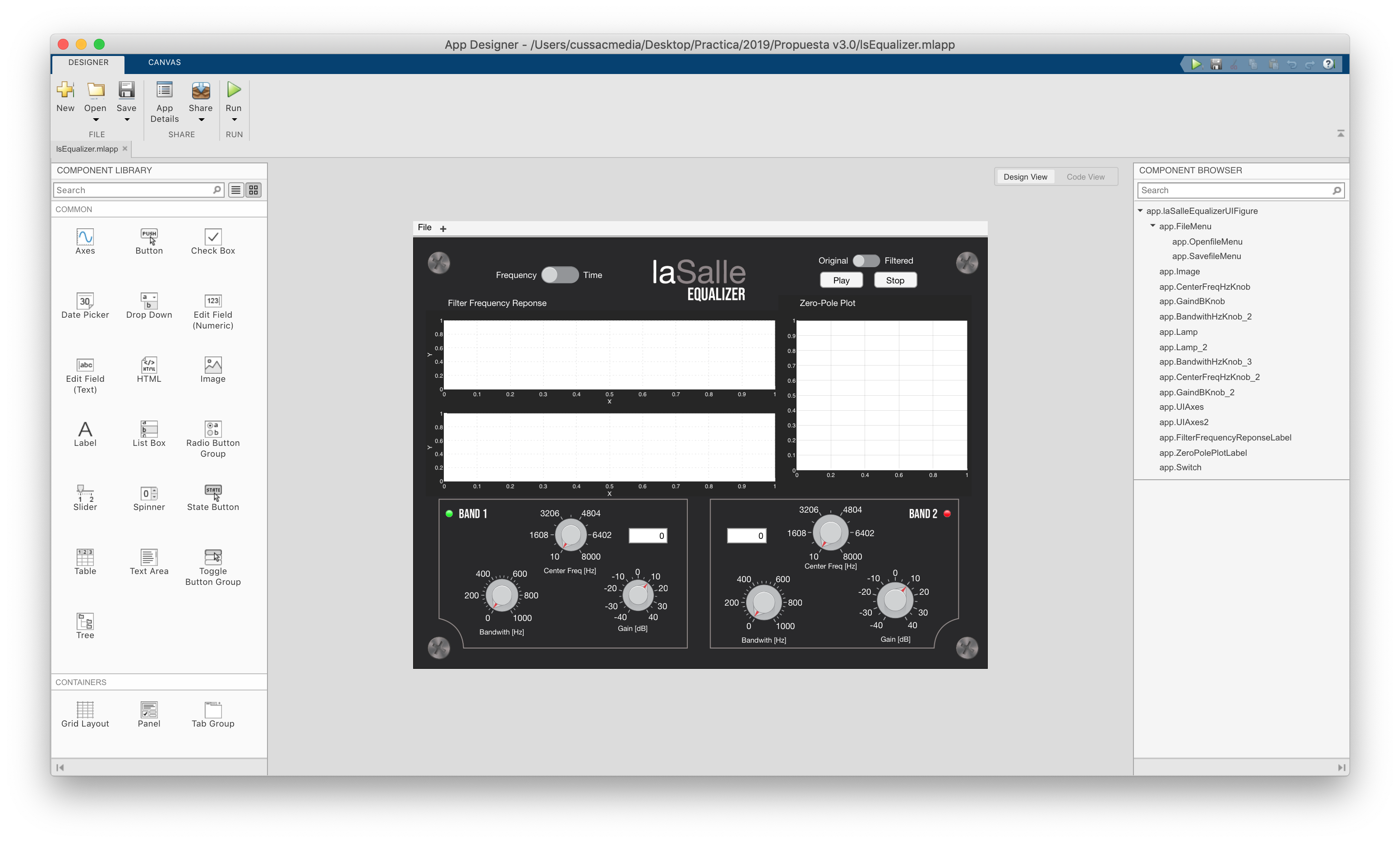
La **segona fase** s’entregarà fins el **dia 14/1/22**.

# 3. INTRODUCCIÓ A APP DESIGNER

App Designer es una nova eina que ofereix el software el software MATLAB i que integra les dos principals funcions que implica la creació d’una app: l’organització dels components visuals i interactius en una interfície gràfica d’ usuari (GUI) i la programació del comportament de la app i de cada component que aquesta inclou. Per tant, es l’entorn recomanat per crear apps en MATLAB.

Per obrir App Designer podem fer-ho de dos formes: des del menú APPS de Matlab seleccionant l’opció Design App, o escrivint ***appdesigner*** des de la línia de comandes.

Il·lustració 1: Entorn de treball de l’App Designer.



Aquesta és la interfície principal de l‘App Designer. En la cantonada superior dreta podem observar que podem treballar de dos maneres diferents:

* **Design View:** Ens permet veure la versió gràfica de la interfície incorporant diferents elements arrossegant els components de la Component Library a la interfície.
* **Code View:** És la vista de programador, a on podem programar el codi que s’executarà a l’interactuar amb els diferents elements de la interfície.

Si està activada la pestanya Design View, podem trobar els següents panells:

1. **Design Editor.** Àrea sobre la qual es distribueixen els diferents elements de control i que constitueix una representació del que es veurà en pantalla a l’executar la interfície.
2. **Imagen que contiene captura de pantalla

   Descripción generada automáticamenteComponent Library.** Llista dels diferents components disponibles pel disseny de la aplicació. Aquests es troben classificats en *common components*, *containers*, *figure tools* i *instrumentation*. Per utilitzar-los, només cal arrossegar els elements elegits per l’aplicació al Design Editor.

Il·lustració 2: Component Library.

Pel cas de la nostra aplicació, ens interessa conèixer els següents components. No obstant, es convida als alumnes a que jugueu amb els altres components que ofereix el programa.

|  |  |  |
| --- | --- | --- |
| **ICONO** | **COMPONENTE** | **DESCRIPCIÓN** |
| Interfaz de usuario gráfica, Aplicación  Descripción generada automáticamente | Axes | Permet crear uns eixos o gràfiques. En el nostre cas utilitzarem una gràfica. |
| Forma  Descripción generada automáticamente | Image | Permet incorporar una imatge. La nostra interfície ja té una imatge de fons. |
|  | Button | Permet incorporar botons e interactuar amb ells. |
| Imagen que contiene objeto  Descripción generada automáticamente | Label | Permet crear una etiqueta amb text. |
| Interfaz de usuario gráfica, Aplicación  Descripción generada automáticamente | Knob | Potenciòmetre que determina el valor d’un paràmetre de control (p.ex. el guany d’una banda de l’equalitzador). En el nostre cas en tindrem 6 per poder modificar els paràmetres de cada efecte (igual que a la fase 1), els knobs han de tenir com a valors extrems els mateixos que apareixen a la il·lustració 1. |
| Imagen que contiene captura de pantalla  Descripción generada automáticamente | Menu Bar | Menú desplegable que ens permet obrir i guardar arxius. |
| Imagen que contiene Texto  Descripción generada automáticamente | Switch | Permet seleccionar entre dos o més opcions. En el nostre cas utilitzarem només dos. |
| Imagen que contiene Código QR  Descripción generada automáticamente | Edit Field | És un camp d’ introducció de caràcters numèrics. |

1. **Imagen que contiene captura de pantalla

   Descripción generada automáticamenteButton Properties.** Finestra amb les propietats de cadascun dels objectes utilitzats en el Design Editor. Un cop seleccionats el components es mostren les característiques (diferents per cada tipus d‘objecte) que poden ser modificades.

Il·lustració 3: Button Properties.

Si està activada la pestanya Code View, podem trobar els següents panells:

1. **Code Editor.** Editor que conté tant el codi generat automàticament després d’afegir objectes com el dels diferents *callbacks*, funcions i propietats implementades. Aquest codi és el que haurem de modificar/ampliar perquè la nostra interfície acabi realitzant les funcionalitats del nostre plugin d’àudio.
2. Imagen que contiene captura de pantalla

   Descripción generada automáticamente**Code Browser**. Llista amb els diferents *callbacks*, funcions d’utilitat i propietats creades en la aplicació.

Il·lustració 4: Code browser.

1. **Toolbar.** Correspon a la barra d’eines. Aquesta es divideix en dos pestanyes. Els principals elements d’aquesta barra d’eines son:

|  |  |  |
| --- | --- | --- |
| **ICONO** | **COMPONENTE** | **DESCRIPCIÓN** |
| Icono  Descripción generada automáticamente | Save | Guarda la aplicació amb la que s’està treballant. |
| Imagen que contiene objeto, botiquín de primeros auxilios, reloj  Descripción generada automáticamente | Callback | Crea un *callback* per un determinat element de la interfície. Es tracta d’un petit codi que s’executarà cada cop que l’usuari modifiqui l’estat de l’element (p.ex. si modifiquem el guany volem veure un canvi en la funció de transferència). |
| Imagen que contiene objeto, cuarto, reloj  Descripción generada automáticamente | Function | Crea una funció d’ utilitat. |
|  | Property | Crea una propietat. |
| Interfaz de usuario gráfica  Descripción generada automáticamente | Run | Executa l’ aplicació. |

A continuació us indiquem el workflow que haureu de complir dins de la aplicació del vostre plugin d’àudio creat amb l’eina App Designer, que seria el següent:

1. Crear la finestra base de la nostra interfície del plugin d’àudio. En el cas de la nostra aplicació, es podria afegir una imatge de fons equalitzador que donés un cert estil (skin) per aquest tipus d’aplicacions d’àudio.
2. Afegir els diferents components a la interfície. Podeu canviar el nom dels components per identificar-los de manera més ràpida quan editeu les *callback function* de l’aplicació..
3. Afegir una funció *callback* per a cada component de la nostra interfície dins del codi del Code Editor que s'executarà cada cop que interactuem amb aquest component.

En l’apartat següent se us explica amb més detall com portar a la pràctica aquest workflow genèric pel cas pràctic del nostre plugin d’àudio.

En el següent vídeo podreu veure un petit tutorial de funcionament de l’App Designer, com a material complementari a aquesta memòria:

[Link al vídeo sobre App Designer](https://lasalleuniversities-my.sharepoint.com/:v:/g/personal/joanclaudi_socoro_salle_url_edu/EQnEgLPIHt1NghBBnEJBGeMBPDvx_Vc1JKLdURkT6zo9yQ)

# FASE II

La Fase II consisteix en la realització d’una interfície gràfica per controlar els efectes d’àudio en temps real del nostre Plugin: equalitzador de tres bandes, eco i reverberació, ja implementats i testejats a la fase I.

Per poder realitzar el funcionament dels efectes a temps real, serà necessari que useu un micròfon i uns auriculars per tal d’evitar que l’àudio que prové del micròfon processat es realimenti a través de l’escolta del mateix senyal. Si no ho feu així, el més probable és que l’àudio acabi saturant degut a aquest efecte de realimentació positiva, que és un efecte bastant desagradable i molest.

Matlab disposa d’una llibreria nativa (funcions audioDeviceReader i audioDeviceWriter) que permet utilitzar la targeta de so de l’ordenador per poder processar l’àudio a temps real (llegir trames del conversor A/D que connecta amb el micròfon del PC i escriure trames cap al conversor D/A que connecta amb el dispositiu de sortida – auriculars connectats al PC en el nostre cas). Degut als alts recursos que necessita el software per realitzar el processament gràfic, es treballarà en dos modes de forma separada, fent ús d’un switch anomenat “onoff”. Per tant, per evitar saturar de càlculs la CPU de l’ordinador, el funcionament que ha de realitzar la aplicació és el següent:

1. Només abans d’activar el processament d’àudio (valor del switch a OFF), s’ha de poder modificar els paràmetres inicials dels efectes (guanys en dB de l’equalitzador, etc.). Un cop activem el processament a temps real (valor del switch a ON) ja no podrem modificar cap dels paràmetres i aquests apareixeran de manera bloquejada fins que apaguem l’efecte (OFF).
2. Per tant, el sistema ha de permetre o bé escoltar l’efecte a temps real o bé veure el senyal de manera gràfica (no tots dos alhora).
   1. En el mode gràfic, s’ha de poder elegir entre visualitzar el senyal temporal o bé la seva representació freqüencial en un eix logarítmic de freqüències en Hz. Per això usarem un altre switch que anomenarem Freq-Time. D’altra banda, tindrem un altre switch addicional que ens permetrà triar si volem veure la representació de la senyal de sortida de l’efecte o bé l‘original, el qual s’anomenarà Origi-Filt.
   2. El segon switch (listen/show), l’utilitzarem per escollir o bé si volem veure la gràfica, o bé escoltar l’àudio amb algun dels efectes aplicats, doncs abans de fer ON hem d’activar o Show o Listen.

Per simplificar la codificació, a continuació es detalla el codi que cal posar en el callback del switch “onoff”, tant quan s’activa la posició de *switch* ON per activar el processament en temps real del so, com per parar el seu processament al prémer el OFF.



Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Del codi anterior, es pot veure que hi ha quatre funcions privades que s’han d’implementar:

1. *blockbuttons(flag) ->* funció que bloqueja o desbloqueja en funció del valor del boolean *flag* tots els botons i interruptors que permeten modificar el filtre o gràfiques. Hi ha un atribut, per cada knob/switch, que fa d’enable de cadascun dels controls de la interfície, l’atribut és el següent, per exemple, knob.enable = true.
2. *processAudio() ->* funció que processa cada trama d’àudio que llegeix el fileReader en funció del efecte seleccionat. Aquesta funció serà molt semblant a la ja implementada a la Fase 2, però amb algunes diferències que se us expliquen més endavant, degudes al fet que estem processant una trama d’un flux continu de àudio, i cal garantir la continuïtat del procés de filtratge digital.
3. *freq\_temp() ->* funció que s’encarregade visualitzar en temps o en freqüència el senyal filtrat o original per cada trama d’àudio.
4. *getCoeficients() ->* funció que s’encarrega d'actualitzar el valor dels coeficients dels efectes o de l’estat dels interruptors, a partir del que l’usuari ha modificat.

Variables globals que s’han de generar:

1. fileReader 🡪 objecte que extreu les trames d’àudio del conversor A/D.
2. deviceWriter 🡪 objecte que envia la trama d’àudio amb efecte cap al conversor D/A.
3. Fs 🡪 freqüència de mostratge dels conversors A/D i D/A.
4. audio 🡪 vector que conté una trama capturada pel conversor A/D.
5. audioEfect 🡪 vector que conté una trama processada amb l’efecte corresponent per a enviar al conversor D/A.

## GUI

Com ja s’ha explicat, l’objectiu serà la creació de una interfície d’ usuari (GUI) que permeti capturar l’àudio del micròfon, aplicar un efecte dels 3 ja treballats a la Fase 1, seleccionar per l’usuari els paràmetres dels efectes i finalment escoltar o visualitzar l’àudio amb l’efecte seleccionat. El disseny de la interfície és lliure, però ha de complir les següents restriccions:

* L'arxiu que crea App Desginer ha de dir-se GUI.mlapp i ha d'estar a la carpeta del projecte amb el resta dels elements.
* Tres ***Discrete Knobs*** o ***Sliders*** que ens permetin ajustar el guany de cadascun dels filtres de l’equalitzador.
* Dos ***Discrete Knobs*** o ***Sliders*** que ajustin elvalor del*delay* i el *echo strength*
* *Un* ***Discrete Knob*** o ***Slider*** que ajusti el temps de reverberació
* Heu de presentar **una única gràfica** (**Axes**). Aquesta ha de mostrar el senyal d'entrada o bé el senyal filtrat, en el domini temporal o bé en el freqüencial. Les gràfiques en el domini freqüencial han de tenir un eix logarítmic en Hz, mentre que la temporal l’eix pot estar en mostres.
* Un ***Switch*** que permeti canviar entre representació en domini temporal o domini freqüencial.
* Un ***Switch*** que permeti seleccionar si volem escoltar l’efecte o veure el senyal en una gràfica
* Un ***Switch*** que ens permeti visualitzar el senyal d'àudio original o el senyal d'àudio amb efecte.
* Les següents funcions privades amb el funcionament anteriorment descrit: ***Blockbuttons(flag),******ProcessAudio()****,****Freq\_temp(),***
* Tots els **callbacks()** corresponents als diferents elements de la interfície i la corresponent interacció.
* Definir una **Property()** privada amb totes les variables globals a emprar.

**NOTA: És important mirar el vídeo sencer d’introducció a l’app designer (adjuntat al final de l’apartat 3) ja que s’expliquen punts molt importants i de gran ajuda, per exemple a partir del minut 5 s’explica com crear els callbacks i propierties.**

Un cop incorporats tots els elements a la interfície, passarem a treballar a nivell de codi. El primer pas serà crear la property privada amb totes les variables globals del sistema, com ara la freqüència de mostreig o les variables on s'emmagatzemen els coeficients dels filtres.

Per accedir a una variable global del sistema o a un element de la interfície, posem sempre la paraula app.(i el nom de la variable) per tenir el seu valor.

Un cop realitzat aquest pas, a la funció privada inicial (function startupFcn(app, val1)) llegir els fitxers .mat generats a la Fase 1 i guardar els coeficients en les variables globals del sistema, ja que així podem accedir a aquestes en tot moment. A més, també cal inicialitzar tots els valors dels elements de la interfície.

Per llegir o modificar el valor d'un element, podem fer servir la comanda app.nom\_element.Value.

A partir d'aquí, cal implementar les funcions privades i els callbacks que permetin una interacció completa amb tots els elements de la interfície.

Captura de pantalla de un celular

Descripción generada automáticamente Interfaz de usuario gráfica

Descripción generada automáticamente

Il·lustració 4: Exemple d’interfície gràfica del nostre plugin d’àudio

Com es pot veure, a la figura esquerra tenim actiu el mode gràfic (primer switch està en mode “show”) amb la gràfica de freqüència (segon switch està en mode “Freq”) del senyal original (tercer switch està en mode “Orig”). El switch amb modes “ON” i “OFF” és el que activa o desactiva el funcionament a temps real.

En canvi, a la figura dreta veiem que el processament a temps real està aturat (“OFF”) amb tots els botons i *knobs* actius per poder modificar els paràmetres dels efectes.

## Processament a temps real

En la Fase I, disposàvem de tot el vector de l’àudio seguit i junt, ja que provenia d’un fitxer .wav emmagatzemat i, per tant, només aplicàvem el filtre de l’efecte un cop sobre tot el fitxer. Ara, en canvi, al tenir un codi que funciona en temps real, quan apliquem l’efecte a cada trama d’àudio haurem de guardar les condicions dels registres de memòria del filtre per a que aquests siguin els mateixos a l’inici del processament de la següent trama, perquè no es produeixi cap discontinuïtat.

Recordem la funció per filtrar de Matlab:

[[y](https://es.mathworks.com/help/matlab/ref/filter.html" \l "bt_vs4t-1-y),[zf](https://es.mathworks.com/help/matlab/ref/filter.html" \l "bt_vs4t-1-zf)] = [filter](https://es.mathworks.com/help/matlab/ref/filter.html)(B,A,x,zi)

A on:

**zi:** són les condicions inicials per als registres interns de memòria del filtre, abans d’aplicar el filtre a la trama, especificades com a vector. La seva longitud ha de ser max(longitud(A),longitud(B))-1.

**zf:** són les condicions finals per als registres interns de memòria del filtre (un cop processada la trama).

Per a la primera trama (inicialització) posarem a zi el valor per defecte amb un vector de retard igual a zeros. A partir de la segona trama farem que les condicions inicials d’una trama (Zi) siguin iguals a les condicions finals de la trama anterior (Zf), que ens haurem d’haver guardat en alguna variable de la interfície. D’aquesta manera podrem filtrar correctament els tres efectes en temps real, sense cap tipus de discontinuïtat a l’àudio de sortida.

# 5. ENTREGA

Com a resultat d'aquesta segona fase de la pràctica cal adjuntar en una carpeta comprimida en un .zip, la qual ha de tenir el següent nom: login1\_login2\_login3.zip, les següents coses:

* **Vídeo.** Enllaç al núvol d’un vídeo on es demostri el correcte funcionament de l’aplicació dissenyada. Per tal afecte, ens haurem d’ajuntar un .txt amb el link al vídeo.
* **Recursos.** Incorporar tot el material extra que heu emprat per al disseny de la interfície (p.ex. imatge de fons, etc.).
* **GUI.mlapp.** Fitxer d'AppDesginer que funcioni sense errors.
* **filters.mat** Fitxers amb els coeficients dels filtres de l’equalitzador.

A l'hora de realitzar la pràctica s'han de tenir en compte les consideracions següents:

- Cal complir totes les restriccions que marca cada apartat de l'enunciat.

- El codi de MATLAB s'ha d'executar per si mateix. No s'accepten codis que s'executin a trossos, o que tinguin errors d’execució.