



TECNOLÓGICO
NACIONAL DE MÉXICO

TECNOLÓGICO NACIONAL DE MÉXICO

INSTITUTO TECNOLÓGICO DE OAXACA

ASIGNATURA: FUNDAMENTOS DE TELECOMUNICACIONES

CATEDRÁTICO: MCC. VALVERDE JARQUÍN REYNA

ALUMNOS:

- GARCÍA GARCÍA JOSÉ ÁNGEL
- SANTIAGO SOLANO DAFNE

GRUPO: ISB

HORA: 12:00 – 13:00

CARRERA: INGENIERÍA EN SISTEMAS
COMPUTACIONALES

REPORTE DEL TRABAJO DE UNIDAD 1

OAXACA DE JUÁREZ, OAX, 12/OCTUBRE/2020

ÍNDICE GENERAL

| | |
|----------------------------------|----|
| ÍNDICE DE IMÁGENES | 2 |
| PLANTEAMIENTO DEL PROBLEMA | 3 |
| DISEÑO DE LA SOLUCIÓN | 3 |
| HERRAMIENTAS UTILIZADAS | 5 |
| CÓDIGO | 5 |
| RESULTADOS OBTENIDOS..... | 21 |
| ENLACE DEL VÍDEO | 27 |
| CONCLUSIÓN | 27 |

ÍNDICE DE IMÁGENES

| | |
|---|----|
| Ilustración 1 - Función senoidal..... | 4 |
| Ilustración 2 - Función senoidal con fase | 4 |
| Ilustración 3 - Codificación unipolar para la señal digital | 4 |
| Ilustración 4 - Interfaz grafica | 21 |
| Ilustración 5 - Señal analógica a graficar | 21 |
| Ilustración 6 - Señal digital a graficar | 21 |
| Ilustración 7 - Resultado en dominio de tiempo..... | 22 |
| Ilustración 8 - Resultado en dominio de frecuencia | 22 |
| Ilustración 9 - Señal analógica con frecuencia 0 a graficar..... | 23 |
| Ilustración 10 - Segunda señal digital a graficar | 23 |
| Ilustración 11 - Resultado de la segunda señal en dominio de tiempo | 23 |
| Ilustración 12 - Resultado de la segunda señal en dominio de frecuencia | 24 |
| Ilustración 13 - Señales con fase | 24 |
| Ilustración 14 - Gráfica de fase de 0 grados..... | 25 |
| Ilustración 15 - Gráfica de fase de 90 grados..... | 25 |
| Ilustración 16 - Gráfica de fase de 180 grados | 26 |
| Ilustración 17 - Gráfica de fase de 270 grados | 26 |

PLANTEAMIENTO DEL PROBLEMA

Para realizar la práctica de esta unidad, el catedrático nos brindó la información necesaria para trabajar, este material es el documento PDF de nombre “Trabajo de unidad” encontrado en la plataforma MOODLE, en ese documento encontramos las instrucciones a seguir para el desarrollo de la práctica y la explicación de cada una de ellas.

El trabajo o problema solicitado, es crear un programa en Matlab que, al introducir los datos necesarios, genere una señal analógica y digital al mismo tiempo, además debe calcular la frecuencia y la tasa de bits respectivamente. El catedrático hace énfasis en que la grafica de la señal analógica se crea a partir del periodo y la amplitud, por otra parte, la digital se crea a partir de un tren de bits.

DISEÑO DE LA SOLUCIÓN

Como ya mencioné anteriormente, el trabajo de esta unidad es básicamente crear un programa por lo tanto se trata de programar, para esto primero tuvimos que investigar sobre la sintaxis de Matlab, aprender sobre este. También complementar la teoría sobre las señales, desde cómo se construyen y cómo se comportan, esto desde buscar la función que nos permitiera graficar una señal analógica y de la construcción de la digital.

Posterior a hacer eso, lo que hicimos fue investigar cómo crear una interfaz gráfica con Matlab, en este caso hicimos uso de la herramienta App Designer de Matlab, ya que nos ayudaba mucho y facilitaba el trabajo para crear una interfaz completa.

Con teoría sobre Matlab, diseño de interfaz y los conocimientos en señales, tocaba empezar a programar. Para esto se empezó con diseñar la interfaz, colocar los componentes en sus respectivas ubicaciones cómo lo solicitaba la práctica.

Con los elementos colocados y la interfaz elaborada, empezamos a programar las acciones de los botones. Que dichas acciones corresponden directamente a la graficación de cada una de las señales, para esto se utilizaron varias funciones que incorpora Matlab y se aplicó la lógica suficiente para resolver el problema planteado.

Las funciones que se emplearon de Matlab para graficar son:

plot: Nos sirve para graficar en 2D y se utilizó para representar la senoidal, específicamente para la gráfica analógica con un periodo diferente de 0 y está representada en el dominio de tiempo.

rectangle: Función de Matlab que nos permite dibujar nuestra señal analógica cuando su periodo es 0 y además está representada en el dominio de tiempo.

bar: Función de Matlab que nos permite hacer un grafico de barra para la señal analógica que está representada en el dominio de frecuencia.

stairs: Función de Matlab que utilizamos para graficar el tren de bits para la señal digital,

Para las señales analógicas implementamos la función senoidal que nos permitiera graficar la señal en base una onda seno, que es la más simple, la función que utilizamos es la siguiente:

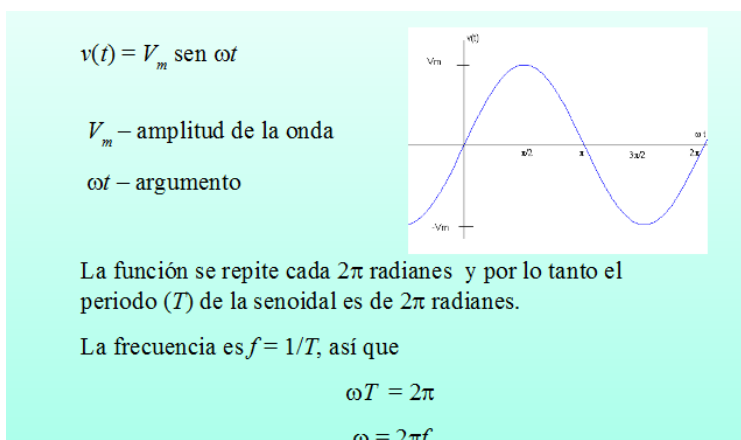


Ilustración 1 - Función senoidal

Aunque realizamos algunos ajustes para implementar esta función, ya que agregamos la fase en radianes para graficar con la fase dada.

Forma general de la senoide $v(t) = V_m \text{ sen } (\omega t + \theta)$

θ – ángulo de fase.

Ilustración 2 - Función senoidal con fase

Por otra parte, la señal digital graficada con una entrada de bits, fue implementada gracias a la codificación unipolar. Además, se usó la función

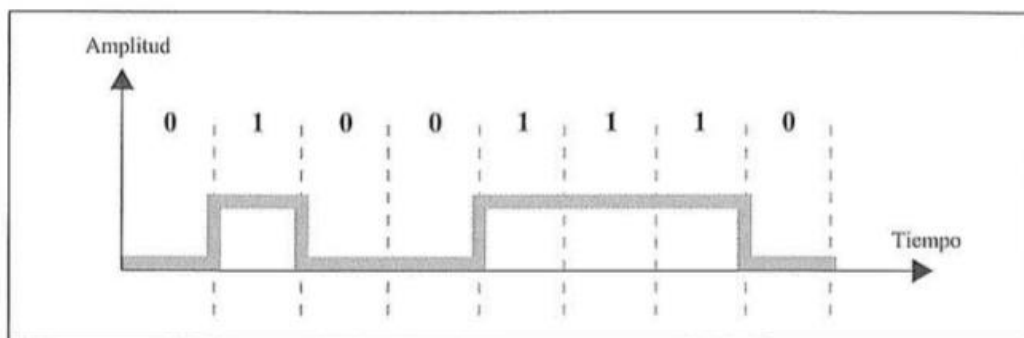


Figura 5.4. Codificación unipolar:

Ilustración 3 - Codificación unipolar para la señal digital

HERRAMIENTAS UTILIZADAS

- Computadora con sistema operativo Windows o donde te permite tener Matlab.
- Matlab versión R2020a
- Bitácora proporcionada

CÓDIGO

```
classdef TrabajoUnidad < matlab.apps.AppBase
```

```
% Properties that correspond to app components
```

```
properties (Access = public)
```

```
    UIFigure          matlab.ui.Figure
```

```
    GridLayout        matlab.ui.container.GridLayout
```

```
    LeftPanel         matlab.ui.container.Panel
```

```
    GARCAGARCAJOSNGELSANTIAGOSOLANODAFNELabel matlab.ui.control.Label
```

```
    RightPanel        matlab.ui.container.Panel
```

```
    GridLayout2       matlab.ui.container.GridLayout
```

```
    UIAxes            matlab.ui.control.UIAxes
```

```
    UIAxes2           matlab.ui.control.UIAxes
```

```
    Panel             matlab.ui.container.Panel
```

```
    PARMETROSDASEALANALGICALabel matlab.ui.control.Label
```

```
    AmplitudVEditFieldLabel matlab.ui.control.Label
```

```
    AmplitudVEditField matlab.ui.control.NumericEditField
```

```
    FaseLabel         matlab.ui.control.Label
```

```
    FaseEditField     matlab.ui.control.NumericEditField
```

```
    FrecuenciaHzEditFieldLabel matlab.ui.control.Label
```

```
    FrecuenciaHzEditField matlab.ui.control.NumericEditField
```

```
    PARMETROSDASEALDIGITALLabel matlab.ui.control.Label
```

```
    GraficarATButton  matlab.ui.control.Button
```

```
    BorrarButton       matlab.ui.control.Button
```

```

TasadebitbpsLabel        matlab.ui.control.Label
TasadebitbpsEditField     matlab.ui.control.NumericEditField
TrendebitsEditFieldLabel  matlab.ui.control.Label
TrendebitsEditField       matlab.ui.control.EditField
IntervalodebitsLabel      matlab.ui.control.Label
IntervalodebitsEditField  matlab.ui.control.NumericEditField
GraficarAFButton         matlab.ui.control.Button
PeriodosEditFieldLabel    matlab.ui.control.Label
PeriodosEditField         matlab.ui.control.EditField
end

```

```

% Properties that correspond to apps with auto-reflow
properties (Access = private)
    onePanelWidth = 576;
end

```

```

% Callbacks that handle component events
methods (Access = private)

```

```

% Button pushed function: GraficarATButton
function GraficarATButtonPushed(app, event)
    % Ajustamos las dos gráficas
    LimpiarGraficas(app, event)

```

```

% GRÁFICA DE SEÑAL ANALÓGICA

```

```

a = app.AmplitudVEditField.Value; % Recuperamos la amplitud en V
per = app.PeriodosEditField.Value; % Recuperamos el periodo en s

```

```

if per == '0' % Verificamos si es una señal analógica con periodo de 0.

```

```

a_2 = a * 2;
app.UIAxes.YLim = [-a_2 a_2];
app.UIAxes.YTick = [-a_2 : 1 : a_2];
app.UIAxes.YTickLabel = { -a_2 : a_2};
a ) ' V'];
titulo = [ 'Señal Analógica con Periodo = ' per ' s   Frecuencia = 0 Hz   Amplitud = ' int2str(
title( app.UIAxes,titulo ); % Colocamos el titulo

rectangle(app.UIAxes,'Position',[0 0 1 a],"FaceColor",'r'); % Graficamos si la señal es
analógica y de periodo 0.

else
    fraccion = strsplit( per, '/' );
    p = str2double( fraccion( 1 ) ) / str2double(fraccion( 2 ) ); % Mapeamos el periodo a
    decimales
    g = app.FaseEditField.Value; % Recuperamos los grados para la fase
    f = 1 / p; % Calculamos la frecuencia en hz
    titulo = [ 'Señal Analógica con Periodo = ' per ' s   Frecuencia = ' int2str( f ) ' Hz   Amplitud
    = ' int2str( a ) ' V   Fase = ' int2str( g ) ' grados' ];
    title( app.UIAxes,titulo ); % Colocamos el titulo
    T = [ 0 : 1 / ( f * 32 ) : 1]; % Vector tiempo
    y = a * sin( 2 * pi * f * T + (g * pi / 180) ); % Función senoidal para la señal

    % Modificamos la grafica para que se vea adaptada a nuestros parametros.
    app.FrecuenciaHzEditField.Value = f; % Mostramos la frecuencia calculada
    app.UIAxes.YLim = [-a a];
    app.UIAxes.YTick = [-a : 1 : a];
    app.UIAxes.YTickLabel = { -a : a};
    plot( app.UIAxes, T, y, '-r'); % Graficamos la señal analógica con periodo diferente de 0
end

% GRAFICA DE SEÑAL DIGITAL
GraficarDigital( app, event);

```



```
end
```

```
% Button pushed function: BorrarButton
```

```
function BorrarCampos(app, event)
```

```
    % Limpiamos los campos de parametros y las graficas
```

```
    LimpiarGraficas(app, event);
```

```
    app.AmplitudVEditField.Value = 0;
```

```
    app.PeriodosEditField.Value = '0';
```

```
    app.FaseEditField.Value = 0;
```

```
    app.FrecuenciaHzEditField.Value = 0;
```

```
    app.TrendebitsEditField.Value = ' ';
```

```
    app.IntervalodebitsEditField.Value = 0;
```

```
    app.TasadebitbpsEditField.Value = 0;
```

```
end
```

```
% Callback function
```

```
function LimpiarGraficas(app, event)
```

```
    % Limpiamos las graficas
```

```
    app.UIAxes.reset;
```

```
    cla( app.UIAxes );
```

```
    title( app.UIAxes, 'Señal Analógica' );
```

```
    xlabel( app.UIAxes, 'Tiempo (s)' );
```

```
    ylabel( app.UIAxes, 'Amplitud (V)' );
```

```
    app.UIAxes2.reset;
```

```
    cla( app.UIAxes2);
```

```
    title( app.UIAxes2, 'Señal Digital' );
```

```
    xlabel( app.UIAxes2, 'Tiempo (s)' );
```

```
    ylabel( app.UIAxes2, 'Amplitud (V)' );
```

end

% Button pushed function: GraficarAFButton

function GraficarAFButtonPushed(app, event)

 % Ajustamos las graficas

 LimpiarGraficas(app, event)

 xlabel(app.UIAxes, 'Frecuencia (hz)');

 ylabel(app.UIAxes, 'Amplitud (V)');

 % GRÁFICA DE SEÑAL ANALÓGICA CON DOMIO DE FRECUENCIA

 a = app.AmplitudVEditField.Value; % Recuperamos la amplitud en V

 per = app.PeriodosEditField.Value; % Recuperamos el periodo en s

 if per ~= '0' % Verificamos si se trata de una señal analógica con periodo de 0

 fraccion = strsplit(per, '/');

 p = str2double(fraccion(1)) / str2double(fraccion(2)); % Mapeamos el periodo a
decimales

 f = 1 / p; % Calculamos la frecuencia en hz

 app.UIAxes.XLim = [0 f + 1];

 app.UIAxes.XTick = [0 : 1 : f + 1];

 app.UIAxes.XTickLabel = { 0 : f + 1 };

 else

 f = 0; % Asignamos frecuencia de 0 si el periodo es 0.

 end

 a_2 = a + 5;

 app.FrecuenciaHzEditField.Value = f; % Mostramos la frecuencia calculada

 titulo = ['Señal Analógica con Periodo = ' per ' s Frecuencia = ' int2str(f) ' Hz Amplitud = '
int2str(a) ' V'];

 title(app.UIAxes,titulo); % Colocamos el titulo

 bar(app.UIAxes, f, a, 'g'); % Graficamos la señal analógica en el dominio de frecuencia

```

app.UIAxes.YLim = [-1 a_2];
app.UIAxes.YTick = [ -1 : 1 : a_2 ];
app.UIAxes.YTickLabel = { -1 : a_2 };
% GRAFICA DE SEÑAL DIGITAL
GraficarDigital( app, event);
end

```

% Callback function

```

function GraficarDigital(app, event)
    % GRAFICA DE SEÑAL DIGITAL
    entrada_tren = app.TrendebitsEditField.Value; % Recuperamos el tren de bits
    elementAdd = [ ' entrada_tren( end ) ];
    entrada_tren_2 = [ entrada_tren elementAdd ];
    tbits = str2num( entrada_tren_2 ); % Mapeamos el string de tren de bits a una matriz de 1 y 0
    ib = 1 / length( strrep( entrada_tren, '1' )); % Calculamos el intervalo de bit
    tsb = 1 / ib; % Calculamos la tasa de bits
    T_2 = [ 0 : ib : 1 ]; % Vector tiempo
    title( app.UIAxes2, [ 'Señal Digital del Tren de Bits [ ' entrada_tren ' ] con Intervalo de bit = '
num2str( ib ) ' s y Tasa de bits = ' int2str( tsb ) ' bps' ] );
    app.IntervalodebitsEditField.Value = ib; % Mostramos el intervalo de bit
    app.TasadebitbpsEditField.Value = tsb; % Mostramos la tasa de bits
    stairs( app.UIAxes2, T_2,tbits, '-m'); % Graficamos la matriz de tren de bits
    app.UIAxes2.YLim = [ -1 2 ];
    app.UIAxes2.YTick = [ -1 : 1 : 2];
    app.UIAxes2.YTickLabel = { -1 : 2 };
end

```

% Changes arrangement of the app based on UIFigure width

```

function updateAppLayout(app, event)
    currentFigureWidth = app.UIFigure.Position(3);

```

```

if(currentFigureWidth <= app.onePanelWidth)
    % Change to a 2x1 grid
    app.GridLayout.RowHeight = {647, 647};
    app.GridLayout.ColumnWidth = {'1x'};
    app.RightPanel.Layout.Row = 2;
    app.RightPanel.Layout.Column = 1;
else
    % Change to a 1x2 grid
    app.GridLayout.RowHeight = {'1x'};
    app.GridLayout.ColumnWidth = {44, '1x'};
    app.RightPanel.Layout.Row = 1;
    app.RightPanel.Layout.Column = 2;
end
end
end

% Component initialization
methods (Access = private)

% Create UIFigure and components
function createComponents(app)

% Create UIFigure and hide until all components are created
app.UIFigure = uifigure('Visible', 'off');
app.UIFigure.AutoSizeChildren = 'off';
app.UIFigure.Position = [100 100 1448 647];
app.UIFigure.Name = 'MATLAB App';
app.UIFigure.SizeChangedFcn = createCallbackFcn(app, @updateAppLayout, true);

```

```

% Create GridLayout
app.GridLayout = uigridlayout(app.UIFigure);
app.GridLayout.ColumnWidth = {44, '1x'};
app.GridLayout.RowHeight = {'1x'};
app.GridLayout.ColumnSpacing = 0;
app.GridLayout.RowSpacing = 0;
app.GridLayout.Padding = [0 0 0 0];
app.GridLayout.Scrollable = 'on';

% Create LeftPanel
app.LeftPanel = uipanel(app.GridLayout);
app.LeftPanel.BackgroundColor = [0.2078 0.7059 0.8314];
app.LeftPanel.Layout.Row = 1;
app.LeftPanel.Layout.Column = 1;
app.LeftPanel.FontAngle = 'italic';

% Create GARCAGARCAJOSNGELSANTIAGOSOLANODAFNELabel
app.GARCAGARCAJOSNGELSANTIAGOSOLANODAFNELabel = uilabel(app.LeftPanel);
app.GARCAGARCAJOSNGELSANTIAGOSOLANODAFNELabel.HorizontalAlignment =
'center';
app.GARCAGARCAJOSNGELSANTIAGOSOLANODAFNELabel.FontWeight = 'bold';
app.GARCAGARCAJOSNGELSANTIAGOSOLANODAFNELabel.FontAngle = 'italic';
app.GARCAGARCAJOSNGELSANTIAGOSOLANODAFNELabel.Position = [4.5 4 37 637];
app.GARCAGARCAJOSNGELSANTIAGOSOLANODAFNELabel.Text = {'G'; 'A'; 'R'; 'C'; 'Í'; 'A';
' '; 'G'; 'A'; 'R'; 'C'; 'Í'; 'A'; ' '; 'J'; 'O'; 'S'; 'É'; ' '; 'Á'; 'N'; 'G'; 'E'; 'L'; '-----'; 'S'; 'A'; 'N'; 'T'; 'I'; 'A'; 'G'; 'O'; ' ';
'S'; 'O'; 'L'; 'A'; 'N'; 'O'; ' '; 'D'; 'A'; 'F'; 'N'; 'E'; ' '};

% Create RightPanel
app.RightPanel = uipanel(app.GridLayout);
app.RightPanel.BackgroundColor = [0.0275 0.4784 0.6902];

```

```

app.RightPanel.Layout.Row = 1;
app.RightPanel.Layout.Column = 2;

% Create GridLayout2
app.GridLayout2 = uigridlayout(app.RightPanel);
app.GridLayout2.RowHeight = {'0.4x', '1x'};
app.GridLayout2.RowSpacing = 14.66666666666667;
app.GridLayout2.Padding = [24.5 14.66666666666667 24.5 14.66666666666667];

% Create UIAxes
app.UIAxes = uiaxes(app.GridLayout2);
title(app.UIAxes, 'Señal Analógica')
xlabel(app.UIAxes, 'Tiempo (s)')
ylabel(app.UIAxes, 'Amplitud (V)')
app.UIAxes.PlotBoxAspectRatio = [1.57435897435897 1 1];
app.UIAxes.FontName = 'Arial';
app.UIAxes.XLim = [0 1];
app.UIAxes.ZLim = [0 1];
app.UIAxes.BoxStyle = 'full';
app.UIAxes.XColor = [0.149 0.149 0.149];
app.UIAxes.XTick = [0 0.2 0.4 0.6 0.8 1];
app.UIAxes.XTickLabel = {'0'; '0.2'; '0.4'; '0.6'; '0.8'; '1'};
app.UIAxes.YColor = [0.149 0.149 0.149];
app.UIAxes.YTick = [0 0.1 0.2 0.4 0.6 0.8 1 2 3];
app.UIAxes.YTickLabel = {'0'; '0.1'; '0.2'; '0.4'; '0.6'; '0.8'; '1'; '2'; '3'};
app.UIAxes.BackgroundColor = [0.9294 0.8314 0.2392];
app.UIAxes.Tag = 'grafica_digital';
app.UIAxes.Layout.Row = 2;
app.UIAxes.Layout.Column = 1;

```

```

% Create UIAxes2
app.UIAxes2 = uiaxes(app.GridLayout2);
title(app.UIAxes2, 'Señal Digital')
xlabel(app.UIAxes2, 'Tiempo(s)')
ylabel(app.UIAxes2, 'Amplitud(V)')
app.UIAxes2.PlotBoxAspectRatio = [1.6505376344086 1 1];
app.UIAxes2.BackgroundColor = [0.9294 0.8314 0.2392];
app.UIAxes2.Layout.Row = 2;
app.UIAxes2.Layout.Column = 2;

% Create Panel
app.Panel = uipanel(app.GridLayout2);
app.Panel.ForegroundColor = [1 0 0];
app.Panel.TitlePosition = 'centertop';
app.Panel.Title = 'FUNDAMENTOS DE TELECOMUNICACIONES -- GRAFICADOR DE
SEÑALES ANALÓGICAS Y DIGITALES';
app.Panel.BackgroundColor = [0.502 0.9686 0.4745];
app.Panel.Layout.Row = 1;
app.Panel.Layout.Column = [1 2];
app.Panel.FontWeight = 'bold';
app.Panel.FontSize = 15;

% Create PARMETROSEALANALGICALLabel
app.PARMETROSEALANALGICALLabel = uilabel(app.Panel);
app.PARMETROSEALANALGICALLabel.HorizontalAlignment = 'center';
app.PARMETROSEALANALGICALLabel.FontSize = 18;
app.PARMETROSEALANALGICALLabel.FontWeight = 'bold';
app.PARMETROSEALANALGICALLabel.FontAngle = 'italic';
app.PARMETROSEALANALGICALLabel.Position = [169 98 419 57];

```

```
app.PARMETROSDELASEALANALGICALabel.Text = 'PARÁMETROS DE LA SEÑAL  
ANALÓGICA';
```

```
% Create AmplitudVEditFieldLabel
```

```
app.AmplitudVEditFieldLabel = uilabel(app.Panel);  
app.AmplitudVEditFieldLabel.HorizontalAlignment = 'right';  
app.AmplitudVEditFieldLabel.FontSize = 14;  
app.AmplitudVEditFieldLabel.Position = [364 64 83 23];  
app.AmplitudVEditFieldLabel.Text = 'Amplitud (V)';
```

```
% Create AmplitudVEditField
```

```
app.AmplitudVEditField = uieditfield(app.Panel, 'numeric');  
app.AmplitudVEditField.Tag = 'etqAmplitud';  
app.AmplitudVEditField.FontSize = 14;  
app.AmplitudVEditField.Position = [462 65 100 22];
```

```
% Create FaseLabel
```

```
app.FaseLabel = uilabel(app.Panel);  
app.FaseLabel.HorizontalAlignment = 'right';  
app.FaseLabel.FontSize = 14;  
app.FaseLabel.Position = [157 21 56 23];  
app.FaseLabel.Text = 'Fase (°)';
```

```
% Create FaseEditField
```

```
app.FaseEditField = uieditfield(app.Panel, 'numeric');  
app.FaseEditField.FontSize = 14;  
app.FaseEditField.Position = [228 22 100 22];
```

```
% Create FrecuenciaHzEditFieldLabel
```

```
app.FrecuenciaHzEditFieldLabel = uilabel(app.Panel);
```



```
app.FrecuenciaHzEditFieldLabel.HorizontalAlignment = 'right';
app.FrecuenciaHzEditFieldLabel.FontSize = 14;
app.FrecuenciaHzEditFieldLabel.Position = [342 19 105 23];
app.FrecuenciaHzEditFieldLabel.Text = 'Frecuencia (Hz)';
```

```
% Create FrecuenciaHzEditField
```

```
app.FrecuenciaHzEditField = uieditfield(app.Panel, 'numeric');
app.FrecuenciaHzEditField.Editable = 'off';
app.FrecuenciaHzEditField.FontSize = 14;
app.FrecuenciaHzEditField.Position = [462 20 100 22];
```

```
% Create PARMETROSEALASEALDIGITALLabel
```

```
app.PARMETROSEALASEALDIGITALLabel = uilabel(app.Panel);
app.PARMETROSEALASEALDIGITALLabel.HorizontalAlignment = 'center';
app.PARMETROSEALASEALDIGITALLabel.FontSize = 18;
app.PARMETROSEALASEALDIGITALLabel.FontWeight = 'bold';
app.PARMETROSEALASEALDIGITALLabel.FontAngle = 'italic';
app.PARMETROSEALASEALDIGITALLabel.Position = [901 98 375 57];
app.PARMETROSEALASEALDIGITALLabel.Text = 'PARÁMETROS DE LA SEÑAL DIGITAL';
```

```
% Create GraficarATButton
```

```
app.GraficarATButton = uibutton(app.Panel, 'push');
app.GraficarATButton.ButtonPushedFcn = createCallbackFcn(app,
@GraficarATButtonPushed, true);
app.GraficarATButton.Tag = 'btnGraficar';
app.GraficarATButton.FontSize = 14;
app.GraficarATButton.FontWeight = 'bold';
app.GraficarATButton.Position = [629 86 95 23];
app.GraficarATButton.Text = 'Graficar A/T';
```

```
% Create BorrarButton
app.BorrarButton = uibutton(app.Panel, 'push');
app.BorrarButton.ButtonPushedFcn = createCallbackFcn(app, @BorrarCampos, true);
app.BorrarButton.FontSize = 14;
app.BorrarButton.FontWeight = 'bold';
app.BorrarButton.Position = [642 17 68 24];
app.BorrarButton.Text = 'Borrar';
```

```
% Create TasadebitbpsLabel
app.TasadebitbpsLabel = uilabel(app.Panel);
app.TasadebitbpsLabel.HorizontalAlignment = 'right';
app.TasadebitbpsLabel.FontSize = 14;
app.TasadebitbpsLabel.Position = [843 19 109 27];
app.TasadebitbpsLabel.Text = 'Tasa de bit (bps)';
```

```
% Create TasadebitbpsEditField
app.TasadebitbpsEditField = uieditfield(app.Panel, 'numeric');
app.TasadebitbpsEditField.Editable = 'off';
app.TasadebitbpsEditField.FontSize = 14;
app.TasadebitbpsEditField.Position = [967 22 100 22];
```

```
% Create TrendebitsEditFieldLabel
app.TrendebitsEditFieldLabel = uilabel(app.Panel);
app.TrendebitsEditFieldLabel.HorizontalAlignment = 'right';
app.TrendebitsEditFieldLabel.FontSize = 14;
app.TrendebitsEditFieldLabel.Position = [988 64 79 22];
app.TrendebitsEditFieldLabel.Text = 'Tren de bits';
```

```
% Create TrendebitsEditField
```

```
app.TrendebitsEditField = uieditfield(app.Panel, 'text');  
app.TrendebitsEditField.FontSize = 14;  
app.TrendebitsEditField.Position = [1082 64 100 22];
```

```
% Create IntervalodebitsLabel
```

```
app.IntervalodebitsLabel = uilabel(app.Panel);  
app.IntervalodebitsLabel.HorizontalAlignment = 'right';  
app.IntervalodebitsLabel.FontSize = 14;  
app.IntervalodebitsLabel.Position = [1082 17 118 29];  
app.IntervalodebitsLabel.Text = 'Intervalo de bit (s)';
```

```
% Create IntervalodebitsEditField
```

```
app.IntervalodebitsEditField = uieditfield(app.Panel, 'numeric');  
app.IntervalodebitsEditField.Editable = 'off';  
app.IntervalodebitsEditField.FontSize = 14;  
app.IntervalodebitsEditField.Position = [1215 22 100 22];
```

```
% Create GraficarAFButton
```

```
app.GraficarAFButton = uibutton(app.Panel, 'push');  
app.GraficarAFButton.ButtonPushedFcn = createCallbackFcn(app,  
@GraficarAFButtonPushed, true);  
app.GraficarAFButton.FontSize = 14;  
app.GraficarAFButton.FontWeight = 'bold';  
app.GraficarAFButton.Position = [629 53 95 23];  
app.GraficarAFButton.Text = 'Graficar A/F';
```

```
% Create PeriodosEditFieldLabel
```

```
app.PeriodosEditFieldLabel = uilabel(app.Panel);  
app.PeriodosEditFieldLabel.HorizontalAlignment = 'right';  
app.PeriodosEditFieldLabel.Position = [148 64 65 22];
```

```

app.PeriodosEditFieldLabel.Text = 'Periodo (s)';

% Create PeriodosEditField
app.PeriodosEditField = uieditfield(app.Panel, 'text');
app.PeriodosEditField.HorizontalAlignment = 'right';
app.PeriodosEditField.Position = [228 64 100 22];
app.PeriodosEditField.Value = '0';

% Show the figure after all components are created
app.UIFigure.Visible = 'on';
end
end

% App creation and deletion
methods (Access = public)

% Construct app
function app = TrabajoUnidad

% Create UIFigure and components
createComponents(app)

% Register the app with App Designer
registerApp(app, app.UIFigure)

if nargin == 0
    clear app
end
end
end

```

```
% Code that executes before app deletion
function delete(app)

    % Delete UIFigure when app is deleted
    delete(app.UIFigure)
end
end
end
```

RESULTADOS OBTENIDOS

La interfaz gráfica que tenemos es la siguiente:

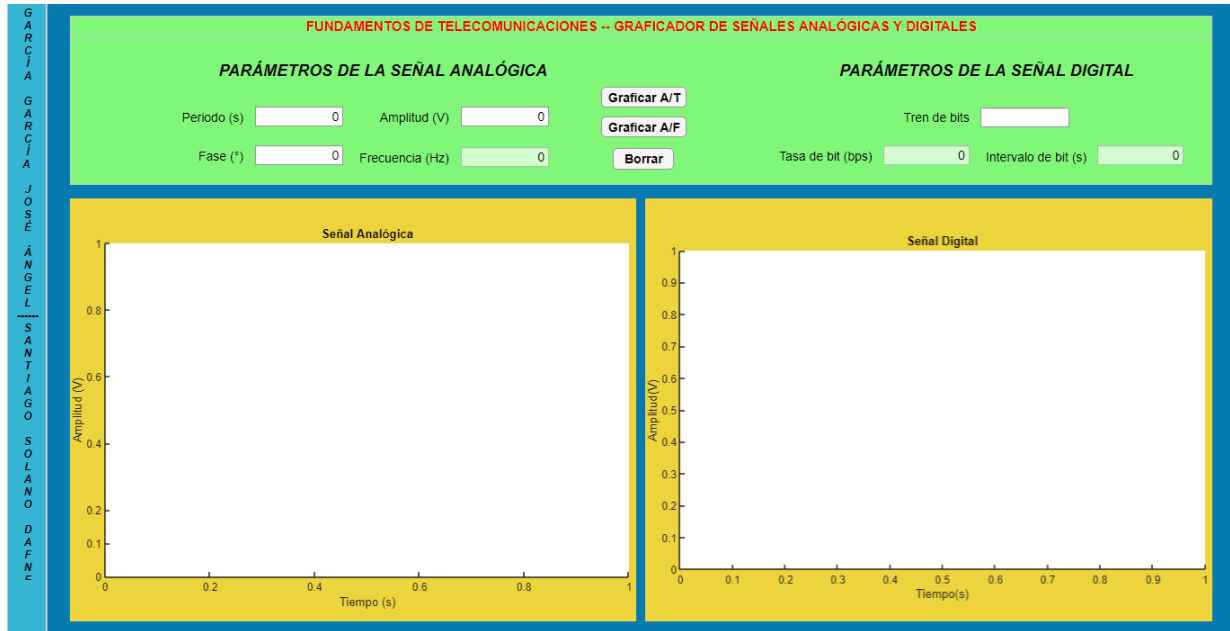


Ilustración 4 - Interfaz grafica

Vamos a hacer la prueba, graficando las siguientes señales

Señal analógica con amplitud de 12V y periodo 1/7 segundos.

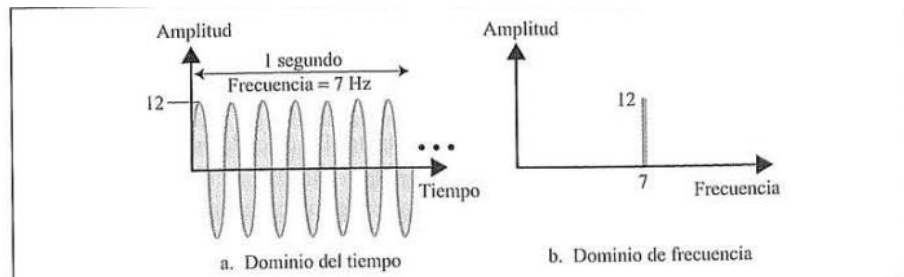


Ilustración 5 - Señal analógica a graficar

Y en la digital graficaremos el tren de bits de [1 0 1 1 0 0 0 1]

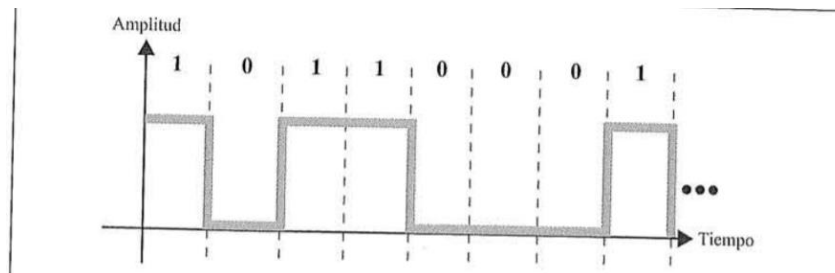


Ilustración 6 - Señal digital a graficar

Introducimos dichos valores y obtenemos.

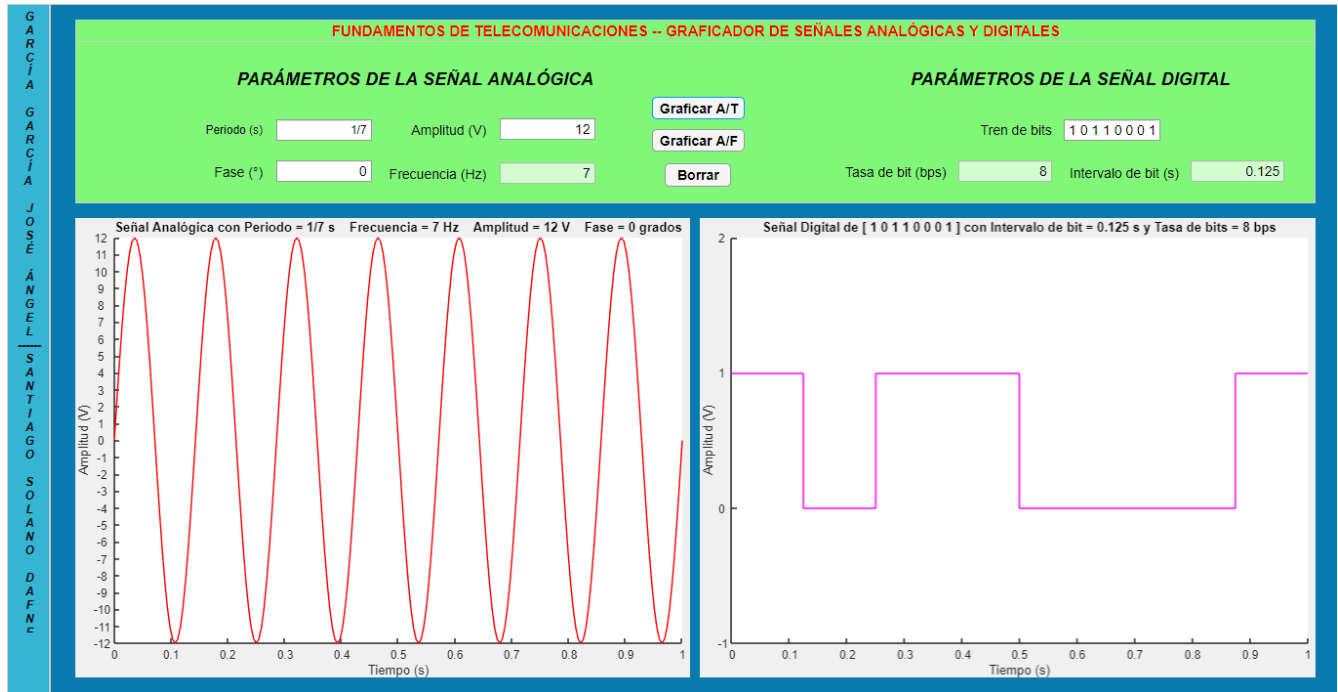


Ilustración 7 - Resultado en dominio de tiempo

Verificamos que realmente si cumplimos con lo solicitado. Además de forma extra podemos colocar a la señal analógica en el dominio de frecuencia.

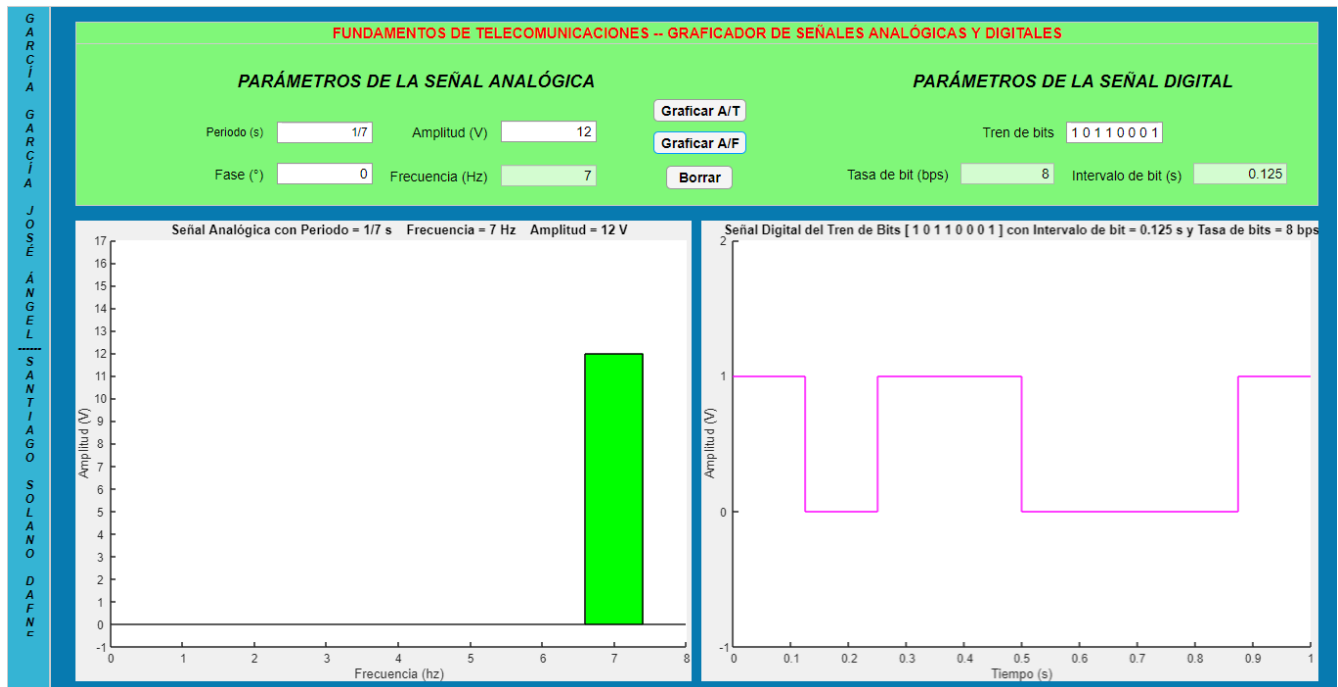


Ilustración 8 - Resultado en dominio de frecuencia

Ahora probamos con una señal analógica continua. Graficaremos la siguiente señal

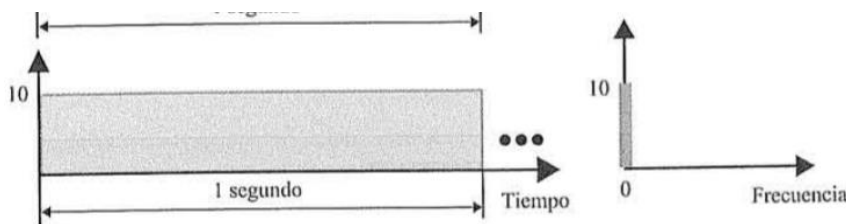


Ilustración 9 - Señal analógica con frecuencia 0 a graficar

Y la digital será:

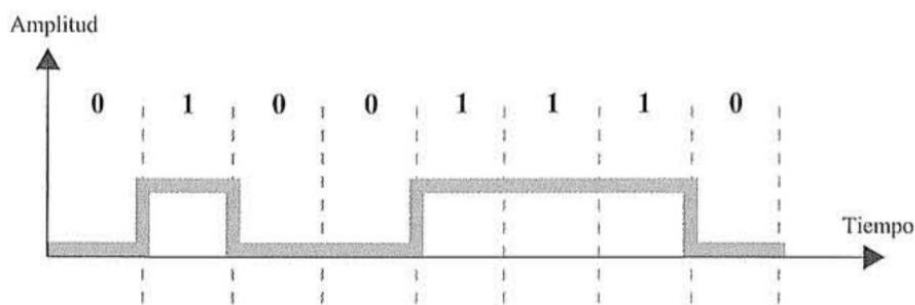


Ilustración 10 - Segunda señal digital a graficar

Nuestro programa, obtiene de resultado en la señal analógica en dominio de tiempo y digital lo siguiente:

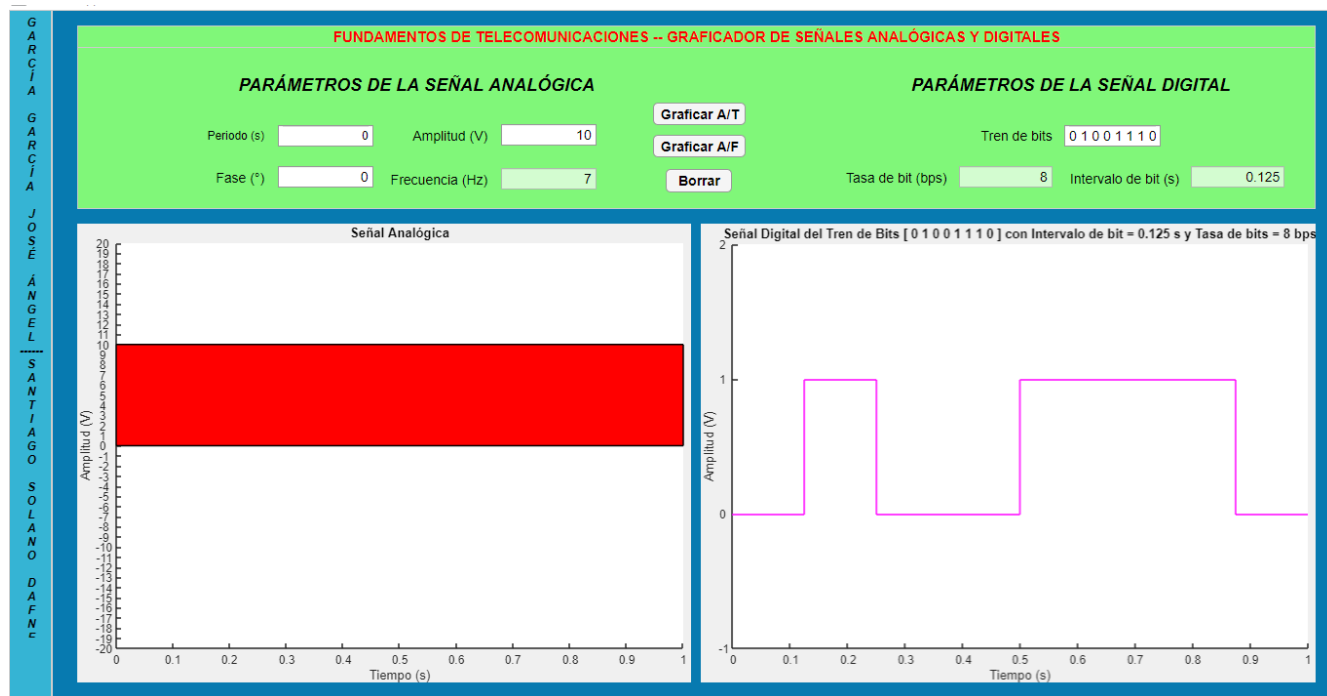


Ilustración 11 - Resultado de la segunda señal en dominio de tiempo

Ahora en la señal analógica en dominio de frecuencia, donde en este caso es 0, se obtiene:

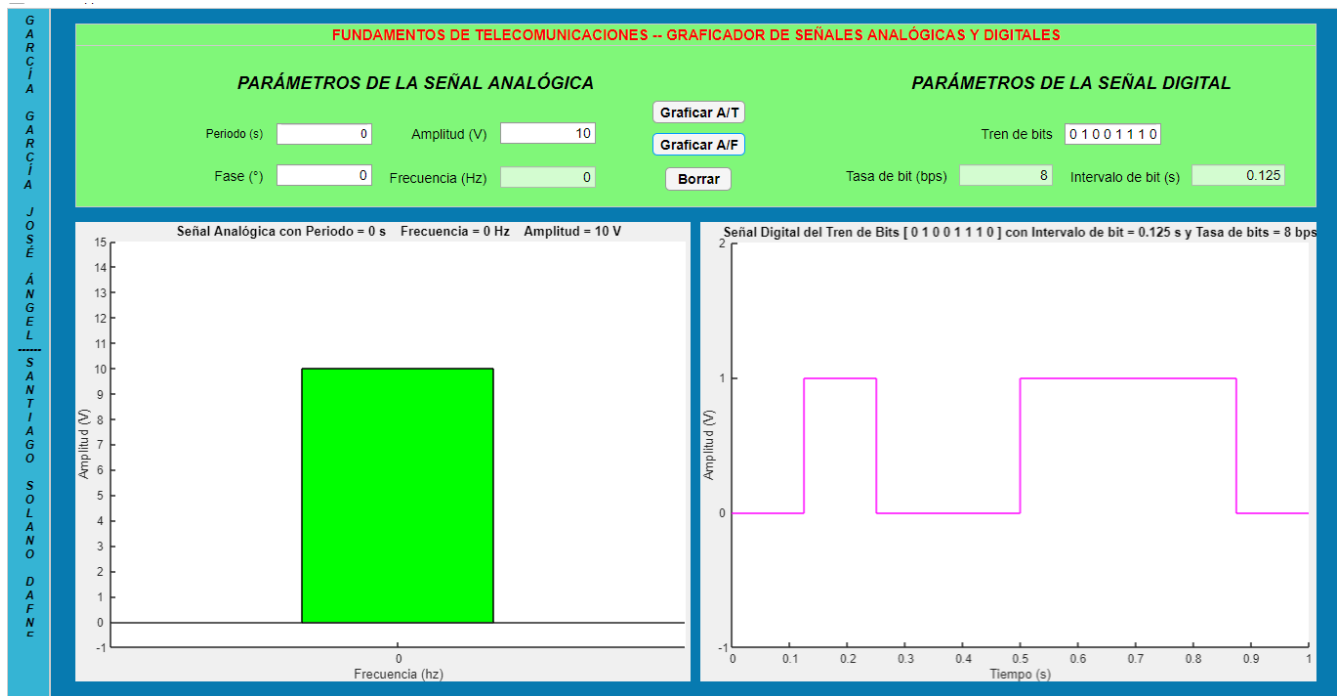


Ilustración 12 - Resultado de la segunda señal en dominio de frecuencia

De forma extra, el programa puede realizar una señal analógica desde una fase indicada, por default es 0 grados, pero esta puede cambiar. Vamos a replicar las siguientes señales analógicas.

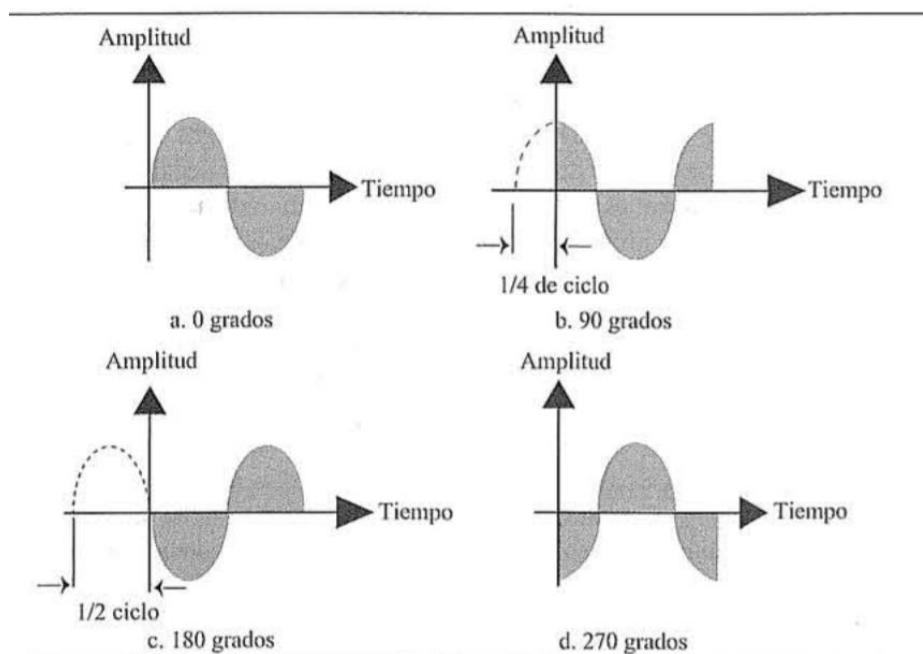


Ilustración 13 - Señales con fase

Fase de 0 grados:



Ilustración 14 - Gráfica de fase de 0 grados

Fase de 90 grados:

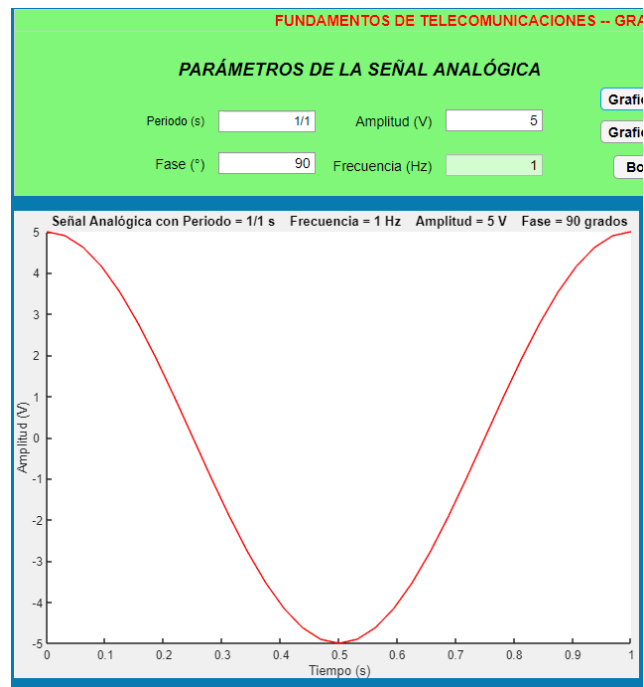


Ilustración 15 - Gráfica de fase de 90 grados

Fase de 180 grados:



Ilustración 16 - Gráfica de fase de 180 grados

Y, por último, con 270 grados:

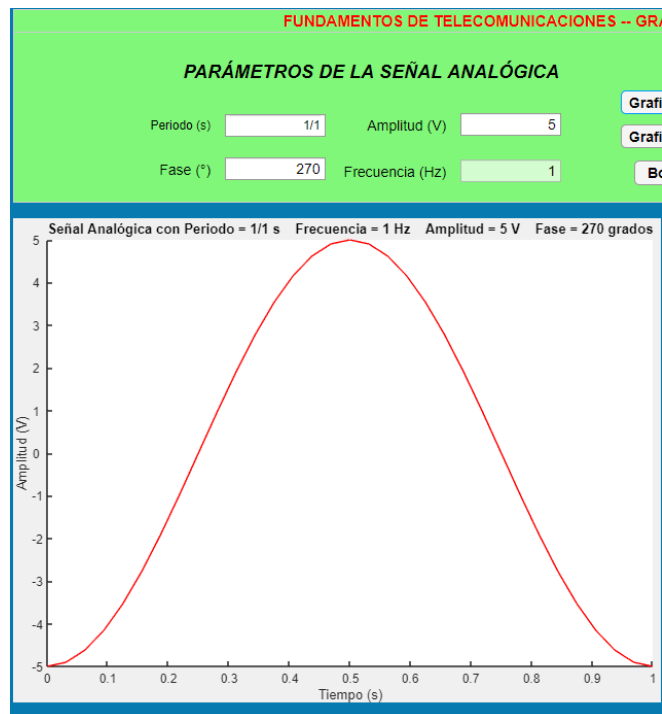


Ilustración 17 - Gráfica de fase de 270 grados

ENLACE DEL VÍDEO

El vídeo en el que se explica a detalle cada una de las fases del programa desde su desarrollo hasta su implementación la puede visualizar en el siguiente enlace:

<https://youtu.be/gZy-88p4g8o>

CONCLUSIÓN

La evidencia que presentamos anteriormente demuestra que en esta práctica se cumplió con el objetivo, que era crear un programa que graficara señales analógicas a partir de un periodo y amplitud dada, así como también que se graficara una señal digital a partir de un tren de bits.

Cómo se observó en el apartado de resultados obtenidos, el programa fue todo un éxito, además de que se le agregaron algunas variantes extras en las señales analógicas, que es el poder graficar en el dominio de frecuencias y aplicar una fase determinada.

Adicional al cumplimiento de este trabajo de unidad, pudimos aprender más sobre lo que son las señales y los componentes de estas. Cabe mencionar que ambos autores de este trabajo teníamos poca experiencia con Matlab, por lo que fue todo un reto el poder completar la práctica, aunque no tan complejo ya que este sistema o lenguaje de programación, nos proporciona muchas herramientas que nos facilitan el trabajo y es por ello, que sólo bastaba con investigar bien algunas de las funciones y aplicarlas a nuestro programa.

Ahora bien, quedando satisfechos con los resultados obtenidos y adquiriendo conocimiento en cuanto al entorno de desarrollo en Matlab esperamos poder seguir adquiriendo aprendizaje por medio de la práctica y una buena enseñanza en el aula virtual.