

Tecnológico de Costa Rica

Escuela de Ingeniería Electrónica



CURSO: EL-5522 TALLER DE COMUNICACIONES ELÉCTRICAS

---

## **Proyecto Final:**

**Modelado de un sistema RF con corrección y detección de errores para una aplicación médica mediante un SoC nRF52832 Nordic Semiconductor**

---

*Profesor:*

Dr.-Ing. Sergio Arriola-Valverde

II Semestre 2025

# Índice

<b>1. Introducción</b>	<b>2</b>
1.1. Aspectos Administrativos . . . . .	2
1.1.1. Conformación de grupos de trabajo . . . . .	2
1.1.2. Foro y Comunicaciones . . . . .	2
1.1.3. Plagio . . . . .	3
1.2. Configuración de Ambiente de Trabajo . . . . .	3
1.3. Control de Versiones en GIT . . . . .	6
<b>2. Motivación</b>	<b>7</b>
<b>3. Descripción de la aplicación</b>	<b>8</b>
<b>4. Descripción del Conjunto de Datos</b>	<b>11</b>
<b>5. Descripción de Requerimientos de Sistema RF</b>	<b>11</b>
5.1. Requerimientos para módulo Transmisor (TX) . . . . .	12
5.1.1. Bloque Latidos por minuto/Prueba Ecostress . . . . .	12
5.1.2. Bloque Preprocesamiento . . . . .	13
5.1.3. Bloque Codificación . . . . .	14
5.1.4. Bloque Modulación . . . . .	15
5.1.5. Bloque Antena . . . . .	15
5.2. Requerimientos del Canal . . . . .	17
5.2.1. Bloque Canal . . . . .	17
5.3. Requerimientos para módulo Receptor (RX) . . . . .	18
5.3.1. Bloque Demodulador . . . . .	18
5.3.2. Bloque Decodificador . . . . .	19
5.3.3. Bloque Visualización . . . . .	20
<b>6. Entregables y Evaluación</b>	<b>21</b>
6.1. Requerimientos para el Flujo de Trabajo . . . . .	21
6.2. Estructura del folder para los entregables . . . . .	22
6.3. Evaluación . . . . .	23
6.4. Entrega del Proyecto . . . . .	24

Esta obra está bajo una licencia [Creative Commons](#) “Reconocimiento-NoCommercial-CompartirIgual 3.0 España”.



# 1. Introducción

Este proyecto final tiene como objetivo principal estudiar algoritmos de detección/corrección de errores, preprocesamiento de señales, esquemas de modulación los cuales son temáticas de estudios del área de comunicaciones eléctricas y serán aplicados a un caso de estudio en el área médica específicamente cuando profesionales en cardiología aplican pruebas de esfuerzo o llamado también prueba de *ecostress*.

No obstante debido a la evolución tecnológica a lo largo de los años muchos algoritmos para la detección/corrección de errores han sido implementados sobre sistemas empujados donde la mezcla de hardware y software se han posicionado en miras de satisfacer condiciones de tiempo real además de requerimientos en algunos casos exigentes a nivel de procesamiento, almacenamiento y ejecución que demanda la industria, sin embargo según la aplicación se han desarrollado circuitos de aplicación específica tales como ASICs.

No obstante mediante la exploración de hardware y software el(la) estudiante en el desarrollo de este proyecto final se enfocará en una aplicación que aborda temáticas en comunicaciones eléctricas, donde deberá usar habilidades desarrolladas en arquitectura de computadores, programación, análisis de antenas, modulaciones digitales y teoría electromagnética para dar solución a la problemática planteada utilizando prácticas de desarrollo que se usan en la industria, para este caso por ejemplo en el área de comunicaciones eléctricas.

## 1.1. Aspectos Administrativos

Antes de ponernos manos a la obra, se discutirán algunos asuntos administrativos que son de suma importancia para una adecuada ejecución del proyecto y a la vez potenciar de manera colaborativa el desarrollo del proyecto final de manera exitosa.

### 1.1.1. Conformación de grupos de trabajo

Se recomienda utilizar la organización del equipo de trabajo inicial definida previamente para laboratorios y exposiciones. En este caso el profesor definirá la cantidad de integrantes en los grupos de trabajo según el cupo de matrícula del curso.

### 1.1.2. Foro y Comunicaciones

Este proyecto se evaluará de forma presencial/remota, por lo que contar con una plataforma online adecuada resulta muy importante para facilitar la comunicación entre alumnos y profesores. Para este caso particular se utilizará la plataforma *Zoom* o *Teams* pero puede variar según las necesidades de el(la) profesor(a).

Para ello, adoptaremos un enfoque de comunidad mediante un **foro** en la plataforma **TecDigital**. En el foro, todos los estudiantes pueden crear nuevos temas a medida que surgen preguntas. Todas las discusiones son públicas para todos los miembros del curso para que cualquier compañero(a) pueda responder y/o agregar más información a la pregunta propuesta. Evite enviar consultas relacionadas con el proyecto directamente al correo electrónico de los profesores, ya que esto evita que otros estudiantes se beneficien de la respuesta. ¡En el foro todos somos un equipo! La única restricción es no compartir código fuente de

trabajo en el foro u otro desarrollo, en cambio, podemos crear discusiones y proponer ideas y conceptos que lleven a la solución. Favor utilizar dicho foro debido a que será de ayuda para todo(as).

### 1.1.3. Plagio

Cualquier evidencia de plagio será investigada a fondo. Si se corrobora, el equipo recibirá una calificación **cero** en el proyecto y se comunicará la incidencia a las autoridades institucionales correspondientes para su posterior procesamiento donde se seguirá el proceso establecido en el artículo 75 del Reglamento de Enseñanza y Aprendizaje del ITCR.

## 1.2. Configuración de Ambiente de Trabajo

El proyecto de manera principal al tener una alta componente de desarrollo de software a continuación se presentan las siguientes recomendaciones y versiones de paquetes en sistemas operativos tanto de Windows, Linux o Mac según sea su computador. A continuación tenga en cuenta lo siguiente con respecto a los paquetes y herramientas de software a utilizar:

### Desarrollo de Software

1. Para el desarrollo de software se deberá utilizar el lenguaje interpretado **Python** preferiblemente en una versión 3.8, es por ello que para evitar problemas a la hora de instalar Python y su interprete en su sistema operativo se recomienda utilizar **Anaconda** versión como mínimo 1.1.10 o superior como ambiente para el desarrollo de software y para la codificación se utilizará **Jupyter Notebook** versión como mínimo versión 6.2.0, no obstante para la descarga de Anaconda ingrese en [acá](#), al instalar el paquete por ejemplo en un ambiente Windows deberá ver un panel frontal como se muestra en la figura 1.

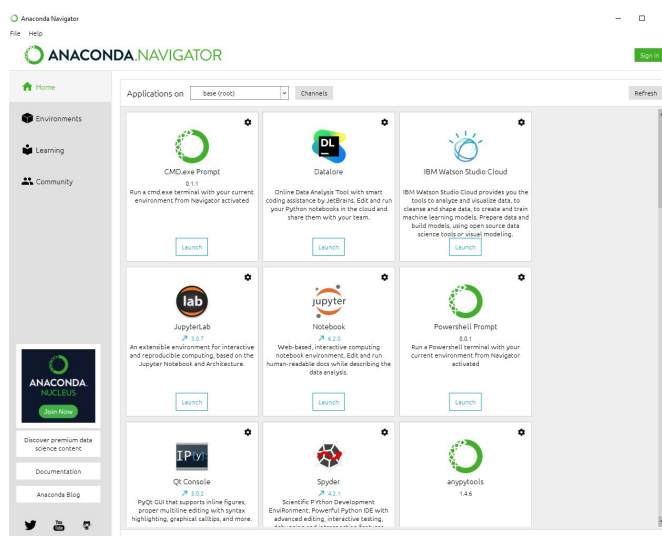


Figura 1: Ventana principal de Anaconda Navigator ejecutado en el sistema operativo Windows.

2. Es recomendable en el ambiente de **Anaconda** instalar los paquetes básicos, para ello utilice la pestaña llamada **Environments** y encontrará un menú de consulta de instalación de paquetes, es por ello que para manejo básico de datos se recomienda instalar los siguientes paquetes:

- Numpy 1.20.1 o superior, si desea más información sobre ejemplos o uso del paquete puede consultar [acá](#).
- Pandas 1.2.2 o superior, si desea más información sobre ejemplos o uso del paquete puede consultar [acá](#).
- Matplotlib 3.3.4 o superior, si desea más información sobre ejemplos o uso del paquete puede consultar [acá](#).

**NOTA:** Cualquier paquete que considere importante para el desarrollo de su proyecto puede ser instalado por cuenta propia, sin embargo garantice que el mismo pueda ser importado de manera adecuada al momento de ejecutar su API en otro computador.

3. Al haber instalado los paquetes básicos mencionados, puede abrir el API de **Jupyter Notebook** dentro del ambiente de **Anaconda Navigator**, al momento de ejecutar el API se abrirá el **Navegador** que tenga por defecto y se mostrará una raíz de archivos tal donde deberá crear o ubicar su folder de trabajo tal como se muestra en la figura 2 .



Figura 2: Raíz de archivos en Jupyter Notebook ejecutado en el sistema operativo Windows.

**NOTA:** En veces resulta necesario cambiar a una dirección los archivos de trabajo para ello si desea más información puede acceder a este [enlace](#).

4. Para crear un archivo **.ipynb** formato de **Jupyter Notebook** ubique en la figura 2 en la parte superior derecha el botón llamado **New** donde deberá escoger el lenguaje interpretado **Python 3**, una vez seleccionado se abrirá un nuevo proyecto el cual por defecto la herramienta le titula como **Untitled** tal como se muestra en la figura 3.

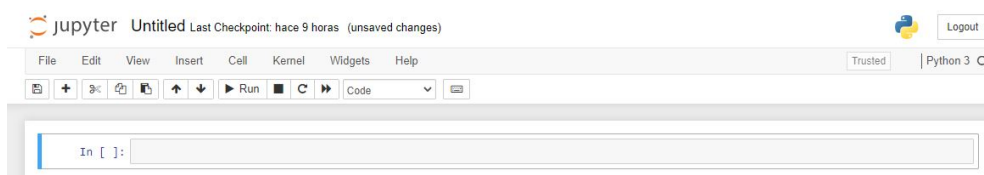


Figura 3: Proyecto en Jupyter Notebook ejecutado en el sistema operativo Windows.

**Nota:** La ventana principal de Jupyter Notebook posee muchas formas y configuraciones que si desea conocer más al respecto se recomienda ver el siguiente video tutorial [acá](#) y complementar con la documentación disponible en el siguiente [enlace](#).

5. Finalmente al crear el nuevo libro en **Jupyter Notebook** puede importar los paquetes instalados en el punto 2 mediante el siguiente código insertado en la celda denominada **In[ ]**:

```
1 # Se cargan las bibliotecas de pandas y numpy
2 import pandas as pd
3 from pandas import ExcelWriter
4 from pandas import ExcelFile
5 import numpy as np
6 import matplotlib.pyplot as plt
7
```

Al insertar el código y haberlo ejecutado mediante el botón de **Run**, la celda deberá ser ejecutada sin que se visualice ningún problema o error según se muestra en la figura 4.

```
In [5]: # Se cargan las bibliotecas de pandas y numpy
import pandas as pd
from pandas import ExcelWriter
from pandas import ExcelFile
import numpy as np
import matplotlib.pyplot as plt
```

Figura 4: Importar paquetes básicos para manejo de datos en Jupyter Notebook ejecutado en el sistema operativo Windows.

**NOTA:** Si tiene algún problema para cargar algún paquete dentro del cuaderno de Jupyter Notebook puede verificar este [enlace](#).

**NOTA:** Recuerde solo por consejo y sugerencia que en Python se debe respetar el orden de indentación, esto es un error frecuente al ejecutar scripts dentro de dicho ambiente debido a que no se verifica el nivel de indentación. Si desea repasar sobre errores de semántica utilice este [enlace](#).

6. Si el paso anterior pudo ser ejecutado sin errores se concluye que su ambiente está listo y los paquetes básicos para manejo de información están correctamente instalados; sin embargo según sus necesidades de desarrollo puede hacer instalación de más paquetes en su ambiente de **Anaconda** utilizando el proceso explicado con anterioridad.

## Herramienta de Prototipado

Como herramienta de software para prototipado se recomienda la herramienta de software libre **Octave** debido al fácil acceso y no pago de licenciamiento o bien si se tiene acceso a **Matlab** es posible usarlo, a continuación se dan detalles de instalación:

1. Para la instalación se recomienda utilizar Octave en su versión 6.1.0 o superior, dicho software tiene instaladores multiplataforma los cuales se pueden encontrar en este [enlace](#).
2. Al instalar la herramienta de Octave dentro de su sistema operativo de preferencia deberá visualizar la ventana principal la cual se muestra en la figura 5.
3. Si no tuvo algún problema durante el proceso de instalación ya puede hacer uso de la herramienta con el licenciamiento gratuito, sin embargo en algunos casos hay que instalar paquetes para ello consulte en este [enlace](#) o bien si desea más información de funciones o toolboxes vea la documentación completa en este [enlace](#).

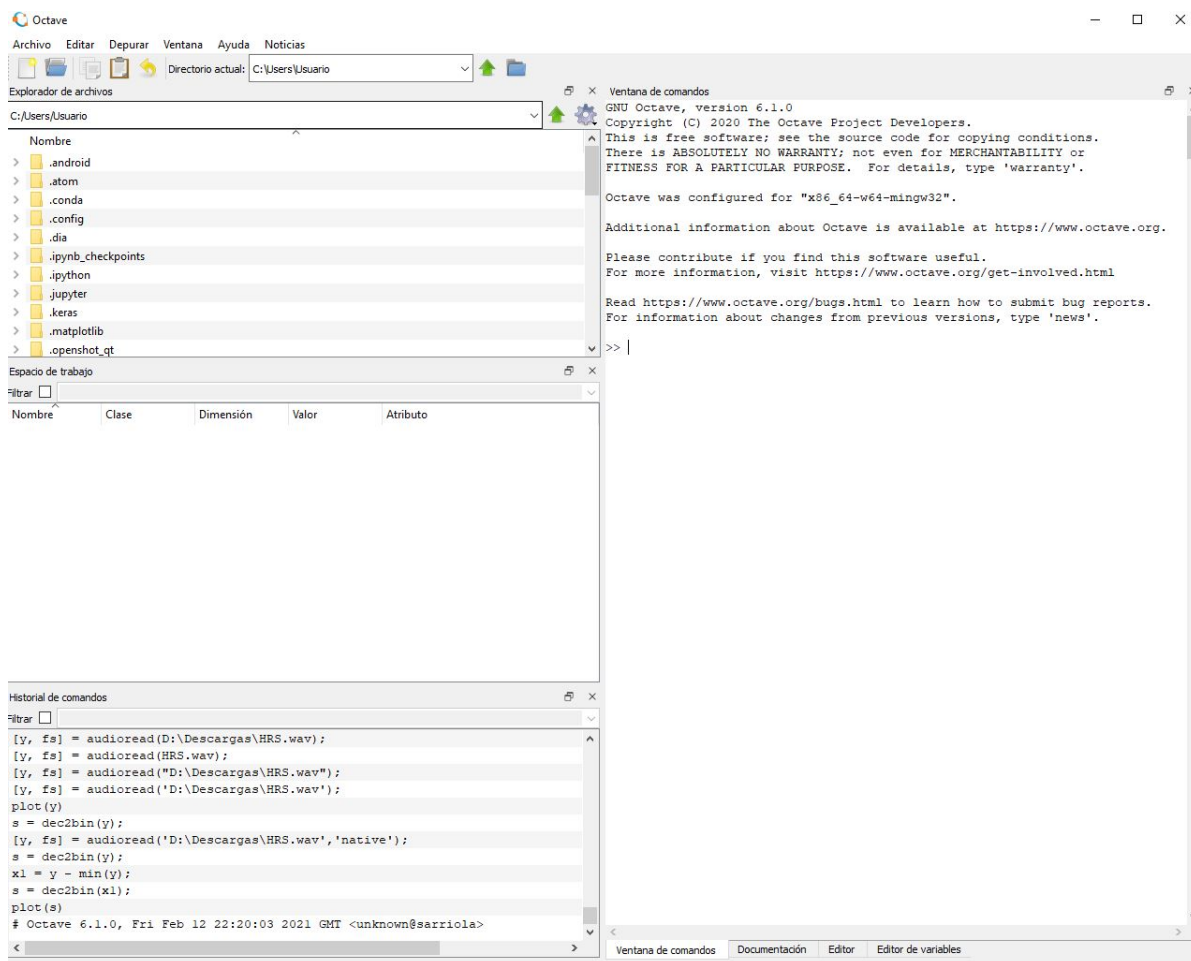


Figura 5: Ventana principal de Octave ejecutado en el sistema operativo Windows.

**NOTA:** Recuerde que entre Matlab y Octave existe cierta compatibilidad de métodos o funciones, no obstante debe tener en cuenta que no todo lo que se desarrolla en Matlab va a ser ejecutado con éxito en Octave es por ello que debe cuidar mucho el desarrollo de prototipado debido a que deberá estar realizado en Octave.

### 1.3. Control de Versiones en GIT

El control de versiones es un concepto poderoso para realizar un seguimiento del progreso de su proyecto, mantener su información segura y trabajar en paralelo junto con los(as) compañeros(as) de su equipo. Por esta razón, el control de versiones se usa ampliamente en todos los proyectos de software serios, incluido el nuestro, para este caso particular usaremos GIT para realizar el desarrollo y tener un seguimiento de nuestro proyecto de forma más sistemática.

Para cumplir con los estándares industriales actuales y facilitar el trabajo paralelo de los estudiantes, todo el proyecto y documentación deberá mantenerse bajo el control de versiones **GIT**.

Todos los documentos asociados con el proyecto (archivos fuente de Jupyter Notebook, archivos de Octave, esquemático y board de Eagle, documentación, etc.) deben mantenerse actualizados en un solo repositorio bajo una jerarquía de ramas. El acceso al repositorio de trabajo **debe** ser otorgado al profesor



para poder clonar sus contenidos y evaluar el trabajo realizado por el equipo. Consulte la Sección 6 para saber qué entregables debe contener su repositorio de GitHub.

Algunos comandos básicos de GIT se discutirán en un tutorial [acá](#) durante la clase para aquellos estudiantes que aún no tienen experiencia con el control de versiones en proyectos de software. Si utiliza el sistema operativo Linux, deberá tener el paquete GIT 2.17.1 o superior **instalado**.

Los siguientes pasos fueron desarrollados en un sistema operativo Linux a nivel de terminal, es por ello que si esta en el sistema operativo Windows podrá instalar el bash de GIT para ello lea y siga este [enlace](#) o bien si desea ver un video tutorial ver este [enlace](#).

Ahora bien en Windows el sistema de control de versiones (SVC) Github ha desarrollado un API de tipo desktop el cual es más sencillo y flexible de usar el cual podrá descargar en este [enlace](#), este enfoque pierde de vista el uso de la terminal el cual no es un proceso recomendable en términos de aprendizaje sin embargo es válido utilizar dicha API en este proyecto.

### 1. Paso 1: Cree una cuenta de GitHub.

Para utilizar GIT, necesitamos un servidor para alojar nuestro proyecto. GitHub proporciona un servidor con cuentas de [gratuitas](#) para equipos. Vaya a [GitHub](#) y cree una cuenta gratuita con su TEC-Email o cuenta personal. **NOTA: Es necesario dejar la cuenta en privado y no pública, esto con la finalidad de evitar copias o réplicas innecesarias del trabajo por terceros.**

### 2. Paso 2: Configure sus Credenciales GIT.

Antes de trabajar con GIT, es necesario ingresar las credenciales de la cuenta creada anteriormente en el host. Para hacerlo, escriba los siguientes comandos en el terminal host sustituyendo su información:

```
1 $ git config --global user.name "YOUR_FIRST_NAME YOUR_LAST_NAME"
2 $ git config --global user.email "YOUR_GIT_ASSOCIATED_EMAIL"
3
```

## 2. Motivación

Debido al desarrollo de múltiples SoC (System-On-Chip) a lo largo de la historia, en el área de telecomunicaciones diversas aplicaciones tales como moduladores, filtros, procesamiento de señales, etc han sido implementados dentro de SoC y MPSoCs debido a que permiten un flujo de desarrollo más granular, controlable, observable y además de la flexibilidad de estos sistemas para cumplir los requerimientos tan exigentes que hoy en día demanda la industria y electrónica de consumo.

En relación a la explosión tecnológica en la actualidad, este proyecto se busca motivar a el(la) estudiante de ingeniería que aplique el diseño de ingeniería y ser capaz de sintetizar un flujo de desarrollo entre software y hardware direccionado a un sistema empotrado enfocándose principalmente en temáticas asociadas a teoría de la información, comunicaciones eléctricas y teoría electromagnética para la concepción de un sistema RF custom que haga un enlace de punto a punto.

No obstante para acercar el desarrollo del proyecto a un entorno realista se utilizará como sistema empotrado objetivo el nRF52832 de Nordic Semiconductor (ver figura 6) el cual posee un SoC ARM Cortex-M4 (Processing System) y una unidad DSP (Coprocessor) es muy recomendable ver la información



detallada del módulo en este [enlace](#) y sobre la arquitectura ARM Cortex-M4 ver este [enlace](#) y para la unidad DSP vea el este [enlace](#).

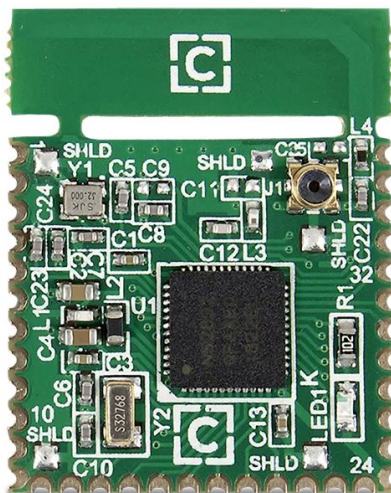


Figura 6: Nordic Semiconductor nRF52832

Es importante recalcar que no podremos usar el hardware por restricciones y disponibilidad del mismo, pero todo el proceso podrá ser prototipado mediante Python, Octave y diseño PCB en Eagle, para luego hacer una estimación de recursos si es posible llevar a cabo la ejecución eventual del sistema prototipo según el sistema empotrado SoC nRF52832, no obstante todo API utilizado en sistemas empotrados generalmente se desarrolla en lenguaje C/C++ preferiblemente en C, sin embargo para este proyecto Python servirá para visualizar la viabilidad de implementación e integración del sistema.

### 3. Descripción de la aplicación

En el área de la salud generalmente el desarrollo de software y hardware debe de cumplir con rigurosos estándares debido a que cualquier proceso anómalo puede generar una afección a la salud de manera irreversible o la muerte, es por ello que todo proceso debe ser ejecutado con la mayor precisión y eficiencia en términos de salvaguardar la integridad de cada persona. Actualmente en cardiología algunos especialistas para conocer y establecer un diagnóstico del estado del corazón generalmente remiten varias pruebas entre ellas la prueba de esfuerzo o bien llamada prueba Ecostress.

La prueba *Ecotress* consiste en someter a un esfuerzo físico donde a su vez mediante un configuración de 12 electrodos se obtiene un electrocardiograma o ECG (a este proceso se le denomina ECG con 12 derivaciones) las cuales están interconectadas a un monitor de ritmo cardíaco donde se visualiza información como presión arterial, ondas QRS y latidos por minutos (bpm). Sin embargo durante dicha prueba un especialista hace un ultrasonido del corazón (Ecocardiograma) para analizar la irrigación sanguínea, apertura de la ventrículos entre otros parámetros importantes. Si desea más información sobre en que consiste la prueba puede ver este [video](#).

La aplicación a desarrollar en este proyecto consiste en el modelado de un sistema prototipo RF (ver figura 7) el cual tendrá como objetivo los siguientes puntos en cada unidad funcional del sistema:

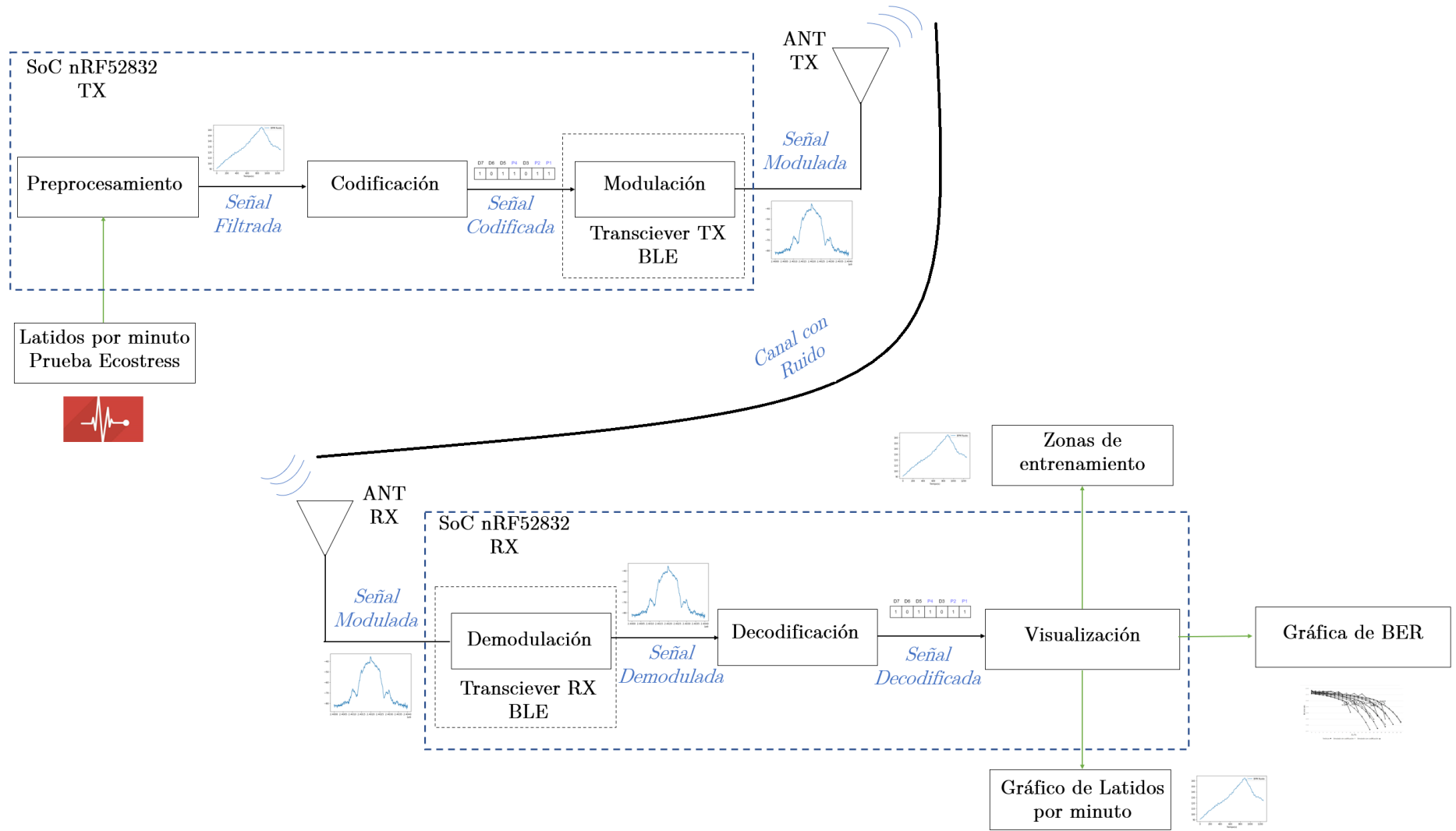


Figura 7: Sistema de Radio Frecuencia a nivel general.

### Transmisor (TX)

- El transmisor (TX) va a tener la capacidad de capturar la información de latidos por minuto (bpm) que provienen de un monitor de ritmo cardíaco, es por ello que asumiremos que los valores son extraídos por una interfaz UART hacia el transmisor (TX) los cuales debido a la interconexión alámbrica entre monitor cardíaco y TX es posible que los datos se vean alterados por ruido (**Los datos para este fin serán dados en una tabla de Excel**).
- Como estrategia previa a codificar la información, es necesario hacer un análisis de los datos en términos de aplicar algún tipo de filtro con el objetivo de filtrar la señal lo mejor posible.
- Como método para asegurar una transmisión correcta de información se implementará un algoritmo de detección/corrección de errores mediante la aplicación de códigos lineales. Estos algoritmos proporcionaran redundancia al sistema lo cual garantizará mayor seguridad en los datos enviados y decodificador por el RX.
- En relación al proceso de modulación, la señal será acondiciona y modulada según las capacidades del SoC para ello se utilizará una tecnología Bluetooth de Bajo Consumo por sus siglas en inglés (BLE) como método de transmisión y recepción, sin embargo para este punto es necesario que el proceso de modulación y demodulación utilizado por BLE sea **modelado** en un sistema de alto nivel con el objetivo de comprender su funcionamiento apriori.

### Antena y Canal

- La antena a utilizar deberá ser propuesta por cada grupo de trabajo y deberá ser embebida dentro del PCB del transceiver (**esquemático y tarjeta serán proporcionados por el profesor**), sin embargo para la selección de la antena se deberá hacer una investigación exhaustiva y revisar parámetros técnicos asociados.
- Debido a que no se posee un modelo exacto del canal, como alternativa el canal será modelado utilizando una fuente de ruido de tipo AWGN, la idea se centra a que el canal dependerá de un SNR donde la información a enviar podrá sufrir algún tipo de alteración y a partir de ello el RX deberá ser capaz de corregir para asegurar la integridad, confiabilidad y visualización al usuario.

### Receptor (RX)

- En relación al proceso de demodulación, la señal deberá ser demodulada según las capacidades del SoC utilizando el mismo esquema BLE, sin embargo para este punto es necesario que el proceso de demodulación utilizado por BLE sea modelado en un sistema de alto nivel con el objetivo de comprender su funcionamiento.
- La unidad de decodificación deberá ser capaz de detectar el error y corregirlo siempre y cuando sea posible, para este punto es necesario tener cotejado la tasa de información con errores y total recibida en términos de determinar la tasa de errores de bits (BER por sus siglas en inglés).
- En el bloque de visualización la información recibida deberá ser estructurada para mostrar información por ejemplo según los latidos por minuto se puede establecer regiones de entrenamiento ver este [enlace](#), además se necesita visualizar la gráfica que latidos por minutos que deberá ser comparada con la gráfica real obtenida por el equipo (**El profesor para el avance 2 les proporcionará esos datos para comparación, el cual deberá ser evaluado con métricas de error**) y finalmente se deberá visualizar un gráfico de BER vrs SNR o bien  $E_b/N_0$  para ello analice la información de este [enlace](#).

## 4. Descripción del Conjunto de Datos

Los datos proporcionados para este proyecto son de un caso de análisis real el cual fue una prueba de Ecostress aplicada en Febrero de 2020 a **una persona de 32 años** (Este dato servirá para determinar zonas de entrenamiento) utilizando bicicleta supina como método de ejercicio, además de un análisis hemodinámico mediante un ecocardiograma completo.

El tiempo de duración de la prueba Ecostress fue de 15 minutos donde fue suspendida por que el paciente sufrió fatiga muscular, aunado a ello se dieron dos intervalos de 4 minutos de descanso y luego 2 minutos adicionales para analizar el periodo de recuperación del paciente. A continuación se detalla la conformación del set datos (**los datos pueden ser descargados en la carpeta llamada Proyectos en TEC-Digital con el nombre Dataset.xlsx**).

- El set de datos posee en total 1260 muestras de pulsaciones por minutos, las cuales fueron tomadas cada 10 segundos. Uno de los retos importantes del proyecto será que el proceso de codificación, envío, recepción y visualización de información **no supere el deadline de 10 segundos**, es por ello que el sistema deberá operar en tiempo real por debajo del deadline establecido.
- Para efectos del proyecto, los datos crudos de latidos por minuto se le adicionó ruido con el objetivo de modelar la conexión alámbrica entre el monitor de ritmo cardíaco y TX. En la figura 8 se muestra la gráfica de latidos por minutos con la fuente de ruido adicionada.

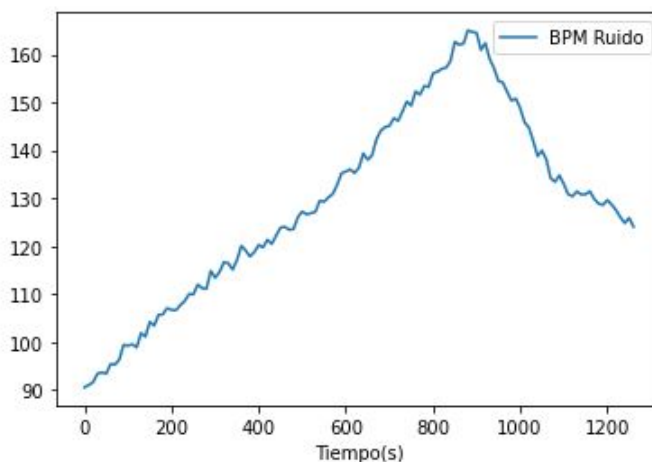


Figura 8: Gráfica de latidos por minutos con adquisición de datos cada 10 segundos y ruido adicionado.

## 5. Descripción de Requerimientos de Sistema RF

En esta sección se describirá de manera puntual los requerimientos de cada bloque funcional que permite el modelo del sistema RF, es por ello que se recomienda que además de los requerimientos planteados cada grupo de trabajo ahonde más sobre cada tema revisando bibliografía pertinente y relevante, es por ello que el profesor a lo largo del curso les brindará una clase acerca de investigación en ingeniería además de consultar el libro de texto [1].

## 5.1. Requerimientos para módulo Transmisor (TX)

### 5.1.1. Bloque Latidos por minuto/Prueba Ecostress

Este bloque tiene como objetivo modelar la adquisición de datos entre el monitor cardiaco y el transmisor (TX) es por ello que los datos a utilizar estarán disponible en un hoja de Excel los cuales están debidamente etiquetados para su uso, para descargar los datos (**pueden ser descargados en la carpeta llamada Proyectos en TEC-Digital con el nombre Dataset.xlsx**) **NOTA: Recuerde que el conjunto de datos posee ruido el cual fue adicionado en términos de dar el acercamiento de una conexión alámbrica con el monitor de ritmo cardiaco y TX.**

Los requerimientos son dados como ruta de desarrollo es por ello que deberán ser respetados, no obstante la revisión será con base al funcionamiento de cada uno o también aspecto de investigación que se requiera. A continuación se detalla que requerimientos se esperan para este bloque:

1. El desarrollo de software deberá ser llevado a cabo en **Python** mediante el apoyo de **Jupyter Notebook**. **NOTA: No se aceptará el desarrollo de software en otro lenguaje, además se recomienda en el libro creado en Jupyter Notebook hacer comentarios de la implementación, para consulta de como comentar código ver este [enlace](#).**
2. El documento de Jupyter Notebook **deberá** estar almacenado en el GitHub respetando la estructura que el profesor le sugiera. **NOTA: Será revisado el aporte de los integrantes de cada grupo en desarrollo y commits que tenga el repositorio.**
3. Se deberá realizar una prueba de normalidad de los datos, para ello se deberá implementar en Python una prueba de normalidad de Anderson-Darling, que le permita obtener métricas tales como el valor p o p-value, desviación estándar y promedio, con base en ello deberá plantear una prueba de hipótesis que permita aceptar o rechazar la normalidad de los datos basado en las métricas mencionadas en conjunto con el intervalo de confianza utilizado. **NOTA: Si tiene interés de refrescar su conocimiento en temas de pruebas de hipótesis, normalidad y estadística inferencial, puede leer este [enlace](#).**
4. Utilizando **Python** implemente un método que permita graficar el histograma de los datos, y con base a ello concluya que tendencia existe para ello, utilice como apoyo la información disponible en este [enlace](#).
5. Para la determinación de datos atípico dentro del conjunto datos, implemente en **Python** un método que le permita graficar lo que se llama en inglés **Box Plot** y además el test de Grubbs, a partir de los resultados obtenidos utilice parámetros de estadística descriptiva y apoyado del **Box Plot** y el test de Grubbs para determinar si hay existencia de datos atípicos. **NOTA: Si desea refrescar sus conocimientos en el tema de datos atípicos puede leer esta información en este [enlace](#).**
6. Finalmente el desarrollo de documentación será muy importante, es por ello que desde ya deberán ir estructurando su documento tipo paper formato **IEEE Transactions** a doble columna en idioma español, para ello pueden usar el template que se encuentra en la carpeta llamada Plantillas LaTeX en TEC-Digital. **NOTA: La documentación es muy importante debido a que tiene un puntaje por cada avance así que no será buena idea no hacerla, y procurar evitar errores gramaticales, ortografía y puntuación debido a que serán penalizados.**

### 5.1.2. Bloque Preprocesamiento

Este bloque tiene como objetivo principal tomar los datos con ruido capturados por la prueba de Ecostress y a partir del análisis estadístico realizado en la sección 5.1.1 se deberá implementar un filtro que permita suprimir el ruido presente en la señal para luego pasar dicha información al bloque de Codificación según el diagrama del sistema mostrado en la figura 7.

Los requerimientos son dados como ruta de desarrollo es por ello que deberán ser respetados, no obstante la revisión será con base al funcionamiento de cada uno o también aspecto de investigación que se requiera. A continuación se detalla que requerimientos se esperan para este bloque:

1. El desarrollo de software deberá ser llevado a cabo en **Python** mediante el apoyo de **Jupyter Notebook**. **NOTA: No se aceptará el desarrollo de software en otro lenguaje, además se recomienda en el libro creado en Jupyter Notebook hacer comentarios de la implementación, para consulta de como comentar código ver este [enlace](#).**
2. El documento de Jupyter Notebook **deberá** estar almacenado en el GitHub respetando la estructura que el profesor le sugiera. **NOTA: Será revisado el aporte de los integrantes de cada grupo en desarrollo y commits que tenga el repositorio.**
3. A partir del análisis estadístico realizado en la Sección 5.1.1, investigue dos posibles técnicas de flitros, para ello deberá establecer una matriz tipo “trade-off” donde pueda definir criterios técnicos que le permita seleccionar el filtro adecuado que permita filtrar ruido del conjunto de datos.
4. El filtro seleccionado deberá ser programado en **Python** y se deberá comparar su funcionamiento mediante un gráfico que permita observar la señal filtrada contra la señal original, en términos de establecer una comparación **cualitativa** del funcionamiento. **NOTA: Puede ver el filtrado de datos en este [enlace](#) y algunas topologías en este [enlace](#).**
5. Se deberá implementar en **Python** un método o función que permita medir el tiempo de ejecución transcurrido al ejecutar el script, para ello se recomienda revisar este [enlace](#) como método de referencia. **NOTA: Se deberá hacer al menos 200 ejecuciones del script para determinar un tiempo promedio de ejecución.**
6. Si el filtro implementado en **Python** se fuera a implementar en un lenguaje de programación de un nivel más bajo por ejemplo C/C++, mediante un proceso metodológico explique la estrategia que le permitiría implementar dicho filtro en el SoC nRF52832. **NOTA: Tome en cuenta para este caso todos los soportes que el fabricante le da en términos de hardware, SDK u otras herramientas que permitan llevar a cabo la programación para dicho SoC. Se recomienda analizar las especificaciones afondo desde el CPU hasta los coprocesadores, ver información en este [enlace](#).**
7. En relación al coprocesador DSP del SoC nRF52832 investigue, que tipo de arquitectura poseen los DSPs y además señale las ventajas de usar este tipo de arquitectura de hardware como coprocesador para aceleración.
8. Finalmente el desarrollo de documentación será muy importante, es por ello que desde ya deberán ir estructurando su documento tipo paper formato **IEEE Transactions** a doble columna en idioma español. **NOTA: La documentación es muy importante debido a que tiene un puntaje por cada avance así que no será buena idea no hacerla, y procurar evitar errores gramaticales, ortografía y puntuación debido a que serán penalizados.**



### 5.1.3. Bloque Codificación

Este bloque tiene como objetivo principal capturar la información filtrada proveniente del bloque de Preprocesamiento, y aplicar un método de codificación de tipo Hamming para ello se deberá estudiar de la temática de Teoría de Información la sección de Códigos Lineales, en este [enlace](#) se muestran algunos aspectos relacionados a sistemas de detección y corrección de errores mediante Hamming, no obstante para un entendimiento mayor se recomienda hacer una búsqueda más profunda en materiales como libros, artículos científicos o tesis.

Los requerimientos son dados como ruta de desarrollo es por ello que deberán ser respetados, no obstante la revisión será con base al funcionamiento de cada uno o también aspecto de investigación que se requiera. A continuación se detalla que requerimientos se esperan para este bloque:

1. El desarrollo de software deberá ser llevado a cabo en **Python** mediante el apoyo de **Jupyter Notebook**. **NOTA: No se aceptará el desarrollo de software en otro lenguaje, además se recomienda en el libro creado en Jupyter Notebook hacer comentarios de la implementación, para consulta de como comentar código ver este [enlace](#).**
2. El documento de Jupyter Notebook **deberá** estar almacenado en el GitHub respetando la estructura que el profesor le sugiera. **NOTA: Será revisado el aporte de los integrantes de cada grupo en desarrollo y commits que tenga el repositorio.**
3. La estrategia de como se hará la lectura y manipulación de la información del conjunto de datos de latidos por minutos **será completamente abierta**, sin embargo el proceso de lectura y manipulación deberá ser explicado con detalle en la documentación (Informe final) además de comentarios en el archivo de **Jupyter**.
4. Se deberá implementar mediante el desarrollo de un algoritmo un sistema de codificación tipo Hamming para ello se deberán usar dos variantes tales como (7,4) y (15,11), para ello tome en cuenta esta información de consulta en el siguiente [enlace](#) además de este [enlace](#). **NOTA: No olvide que no se aceptará desarrollos de software en otro lenguaje de programación.**
5. Una vez implementado el sistema de codificación según los enfoques (7,4) y (15,11) según los resultados que obtenga en cuando a ejecución u otros factores que considere para la evaluación del algoritmo, se deberá formular un tabla comparativa que permita establecer y entender ventajas o desventajas que existen entre los enfoques implementados.
6. Se deberá implementar en **Python** un método o función que permita medir el tiempo de ejecución transcurrido al ejecutar el script, para ello se recomienda revisar este [enlace](#) como método de referencia. **NOTA: Se deberá hacer al menos 200 ejecuciones del script para determinar un tiempo promedio de ejecución.**
7. En relación al coprocesador DSP del SoC nRF52832 determine si conviene acelerar el proceso de cálculo de codificación mediante el uso del coprocesador DSP. **NOTA: Para este apartado es importante una buena argumentación con buen contenido técnico que permita respaldar la decisión o estrategia que cada grupo plantee.**
8. Finalmente el desarrollo de documentación será muy importante, es por ello que desde ya deberán ir estructurando su documento tipo paper formato **IEEE Transactions** a doble columna en idioma español. **NOTA: La documentación es muy importante debido a que tiene un puntaje por cada avance así que no será buena idea no hacerla, y procurar evitar errores gramaticales, ortografía y puntuación debido a que serán penalizados.**



#### 5.1.4. Bloque Modulación

Este bloque tiene con función principal tomar la información proveniente del bloque de Codificación y aplicar el proceso de modulación que emplea la tecnología BLE. Es importante tomar en cuenta que el SoC nRF52832 posee el sistema de modulación establecido mediante una unidad de hardware es por ello que para este apartado **no será necesario desarrollar** el esquema de modulación en Python, sino que el esquema de modulación será desarrollado en alto nivel mediante el apoyo de Octave o Matlab, en términos de comprender el funcionamiento de dicho esquema de modulación.

Los requerimientos son dados como ruta de desarrollo es por ello que deberán ser respetados, no obstante la revisión será con base al funcionamiento de cada uno o también aspecto de investigación que se requiera. A continuación se detalla que requerimientos se esperan para este bloque:

1. El desarrollo de software deberá ser llevado a cabo en **Octave o Matlab** y será posible utilizar toolboxes u otras funciones que se necesiten. **NOTA: No se aceptará el desarrollo de software en otro lenguaje para consulta sobre documentación se puede consultar este [enlace](#).**
2. El script de Octave o Matlab **deberá** estar almacenado en el GitHub respetando la estructura que el profesor le sugiera. **NOTA: Será revisado el aporte de los integrantes de cada grupo en desarrollo y commits que tenga el repositorio.**
3. Investigue e implemente el esquema de modulación que utiliza la tecnología de transmisión BLE. Para la implementación puede extraer información en este [enlace](#).
4. Se deberá implementar en **Octave o Matlab** un método o función que permita medir el tiempo de ejecución transcurrido al ejecutar el script, para ello se recomienda revisar este [enlace](#) como método de referencia.
5. Con base al esquema de modulación utilizado por la tecnología BLE, investigue bajo que estándar IEEE se encuentra regulado y además discuta al respecto de familias o categorías de protocolos de comunicación que pueden utilizarse para la tecnología BLE.
6. Finalmente el desarrollo de documentación será muy importante, es por ello que desde ya deberán ir estructurando su documento tipo paper formato **IEEE Transactions** a doble columna en idioma español. **NOTA: La documentación es muy importante debido a que tiene un puntaje por cada avance así que no será buena idea no hacerla, y procurar evitar errores gramaticales, ortografía y puntuación debido a que serán penalizados.**

#### 5.1.5. Bloque Antena

Este bloque tiene como función enviar el mensaje de información modulado hacia el receptor mediante un canal de comunicación, debido a ello el sistema de antena en el SoC nRF52832 esta basado en tecnología microstripline utilizando un topología tipo *patch*, es por ello que se deberá sugerir una topología de antena y completar el *board* en el software Eagle según las especificaciones técnicas de la banda de transmisión de la tecnología BLE.

Para completar el PCB se puede acceder al software Eagle al Pool de Aplicaciones de la Escuela de Ingeniería Electrónica.

Los requerimientos son dados como ruta de desarrollo es por ello que deberán ser respetados, no obstante la revisión será con base al funcionamiento de cada uno o también aspecto de investigación que se

requiera. A continuación se detalla que requerimientos se esperan para este bloque:

1. Descargue el PCB en en la carpeta de **Proyectos de TEC-Digital llamado PCB.zip** y ejecútelo en el programa de Eagle, al ejecutarlo en Eagle se deberá analizar el número capas que contiene, que tipo de vías esta utilizando, asignación de planos de potencia (VDD y GND), el stack-up utilizado, dimensiones de pads, cables y vías. **NOTA: Es posible que necesite agregar las librerías de Nordic Semiconductor, si desea saber como instalar librerías externas se recomienda leer esta información en este [enlace](#).**
2. En el PCB hay un elemento llamado MM8130-2600 comente la funcionalidad que tiene dicho dispositivo además de las características técnicas que posee dicho elemento y que tipo de elementos se puede conectar en dicho dispositivo.
3. Para la selección de la antenna a desarrollar en Eagle se recomienda revisar las bandas de funcionamiento de la tecnología BLE y posterior a ello se deberá establecer al menos dos topologías de antenna tipo patch y con base en una matriz “trade-off” establezca mediante métricas de diseño cual antenna es la mejor para implementación. **NOTA: Se recomienda consultar la información presente en el siguiente [enlace](#), la cual contiene una serie de antenas tipo patch para la tecnología BLE.**
4. La antenna a implementar deberá ser interconectada en un lugar específico, para ello en la figura 9 se muestra el conector y además de la posible ubicación de la antenna. No obstante si desea ver diversas topologías de antenas con sus respectivas vistas a nivel de PCB puede acceder en este [enlace](#) y puede *layouts* de antenas tipo patch. **NOTA: Recuerde que si selecciona una antenna que se sale de la tarjeta, puede ampliar las dimensiones de la tarjeta pero recuerde que al aumentar las dimensiones el costo de fabricación aumenta.**

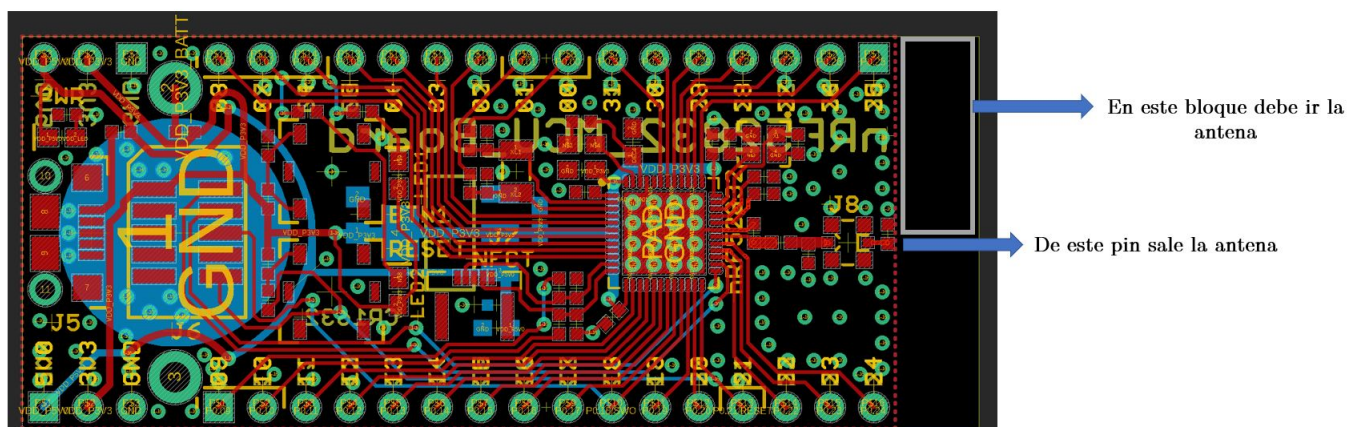


Figura 9: Ubicación de conector de antenna y zona de diseño dentro del software Eagle.

5. Con base a la antenna implementada mediante una revisión bibliográfica relevante localice los patrones de radiación de la antenna además de que software de simulación se puede utilizar para llevar acabo la simulación de la antenna, comente además cual es el objetivo de que la antenna posea un plano a tierra debajo de la capa TOP en el PCB y como debería visualizar el parámetro  $S_{11}$  si se hiciera la medición de laboratorio de la antenna en un analizador vectorial de redes (VNA).
6. Investigue que implicaciones habrían en el patrón de radiación si la antenna tuviera sustrato flexible, para ello puede consultar este artículo científico como base teórica [revisar](#). **NOTA: Recordar que**

el TEC tiene la base de datos IEEE de manera gratuita y abierta usando los credenciales del LAIMI.

7. Establezca que diferencias existen si se utilizan diferentes stack-ups tales como FR-4 o Rogers Materials para el desarrollo de antenas de tipo microstripline. Basado en la tecnología BLE cual sustrato seleccionaría para llevar acabo la fabricación. **NOTA: La argumentación que se proporcione deberá tener contenido técnico conciso y breve, evite frases coloquiales y cualitativas.**
8. Finalmente el desarrollo de documentación será muy importante, es por ello que desde ya deberán ir estructurando su documento tipo paper formato **IEEE Transactions** a doble columna en idioma español. **NOTA: La documentación es muy importante debido a que tiene un puntaje por cada avance así que no será buena idea no hacerla, y procurar evitar errores gramaticales, ortografía y puntuación debido a que serán penalizados.**

## 5.2. Requerimientos del Canal

### 5.2.1. Bloque Canal

Este bloque tiene como función principal modelar de una manera aproximada el canal de comunicación, esto se debe a que no se cuenta con un modelo preestablecido de canal ni mucho menos un modelo de parámetros S, es por ello que se utilizará un ruido de tipo AGWN para llevar a cabo el modelado. Es recomendable hacer lectura de esta información en este [enlace](#) para el entendimiento de los alcances que tiene un modelo de ruido AWGN aplicado en este proyecto y además podrá utilizar el paquete de **Python** llamado *Komm* el cual podrá descargar de este [enlace](#).

Los requerimientos son dados como ruta de desarrollo es por ello que deberán ser respetados, no obstante la revisión será con base al funcionamiento de cada uno o también aspecto de investigación que se requiera. A continuación se detalla que requerimientos se esperan para este bloque:

1. El desarrollo de software deberá ser llevado a cabo en **Python** mediante el apoyo de **Jupyter Notebook**. **NOTA: No se aceptará el desarrollo de software en otro lenguaje, además se recomienda en el libro creado en Jupyter Notebook hacer comentarios de la implementación, para consulta de como comentar código ver este [enlace](#).**
2. El documento de Jupyter Notebook **deberá** estar almacenado en el GitHub respetando la estructura que el profesor le sugiera. **NOTA: Será revisado el aporte de los integrantes de cada grupo en desarrollo y commits que tenga el repositorio.**
3. Para el desarrollo de software se recomienda que haga análisis del código de C en este [enlace](#) y en el código llamado AWGN.c en la carpeta llamada Proyectos en TEC-Digital y explique en detalle sobre el método Box-Muller. **NOTA: Puede hacer lectura sobre el método Box-Muller en este [enlace](#).**
4. Para garantizar un parámetro adecuado de aleatoriedad en los datos generados por cada ejecución, se deberá alternar el valor de semilla, para ello consulte la información que se muestre en este [enlace](#). Recuerde que el modelo del canal mediante AWGN puede ser utilizado el toolbox de *Komm*.
5. Investigue a que se debe el *Fading* en canales de comunicación, y además explique como se vería afectado el enlace de comunicación utilizando tecnología BLE, si desea más información puede

consultar este [enlace](#).

6. Con base a la frecuencia de operación de la tecnología BLE si al enviar la información por el canal de comunicación existe el efecto llamado *Multipath*, explique en que consiste este efecto y que afectación puede haber en el enlace de comunicación y plantee además una estrategia para minimizar este efecto, puede ver más información en este [enlace](#).
7. A partir de la información a enviar por el canal de comunicación, se deberá investigar y determinar de manera **cuantitativa** la capacidad del canal y además en relación a la velocidad de transferencia de datos que posee el SoC nRF52832 analice si el canal se puede saturar durante la transmisión de la información, para ello se recomienda leer este [artículo](#) que puede servir como ayuda.
8. Si el canal tuviera la capacidad de realizar transferencias de datos en alta velocidad, investigue como se vería afectado el diagrama de ojo en términos de SNR y jitter. Para complementar la información puede hacer leer esta información en este [enlace](#).
9. Finalmente el desarrollo de documentación será muy importante, es por ello que desde ya deberán ir estructurando su documento tipo paper formato **IEEE Transactions** a doble columna en idioma español. **NOTA: La documentación es muy importante debido a que tiene un puntaje por cada avance así que no será buena idea no hacerla, y procurar evitar errores gramaticales, ortografía y puntuación debido a que serán penalizados.**

## 5.3. Requerimientos para módulo Receptor (RX)

### 5.3.1. Bloque Demodulador

Este bloque tiene como función demodular y extraer la información que fue transmitida a lo largo del canal, no obstante es importante acotar que al igual que el modulador este bloque no será codificado en Python sino en Octave/Matlab debido a que lo que se busca es entender el esquema de modulación y demodulación que posee el SoC nRF52832, es por ello que la implementación será a nivel de comportamiento.

Los requerimientos son dados como ruta de desarrollo es por ello que deberán ser respetados, no obstante la revisión será con base al funcionamiento de cada uno o también aspecto de investigación que se requiera. A continuación se detalla que requerimientos se esperan para este bloque:

1. El desarrollo de software deberá ser llevado a cabo en **Octave/Matlab** y será posible utilizar tool-boxes u otras funciones que se necesiten. **NOTA: No se aceptará el desarrollo de software en otro lenguaje para consulta sobre documentación se puede consultar este [enlace](#).**
2. El script de Octave/Matlab **deberá** estar almacenado en el GitHub respetando la estructura que el profesor le sugiera. **NOTA: Será revisado el aporte de los integrantes de cada grupo en desarrollo y commits que tenga el repositorio.**
3. Investigue e implemente que esquema de demodulación utiliza el BLE. Para la implementación puede extraer información en este para más información puede consultar en este [enlace](#).
4. Se deberá implementar en **Octave** un método o función que permita medir el tiempo de ejecución transcurrido al ejecutar el script, para ello se recomienda revisar este [enlace](#) como método de referencia.

5. Investigue la relación que existe entre el proceso de demodulación FSK contra GFSK, puede utilizar al menos como base este [video](#) y puede complementar con este otro [video](#).
6. Finalmente el desarrollo de documentación será muy importante, es por ello que desde ya deberán ir estructurando su documento tipo paper formato **IEEE Transactions** a doble columna en idioma español. **NOTA: La documentación es muy importante debido a que tiene un puntaje por cada avance así que no será buena idea no hacerla, y procurar evitar errores gramaticales, ortografía y puntuación debido a que serán penalizados.**

### 5.3.2. Bloque Decodificador

Este bloque deberá implementar un algoritmo usando Python que permita llevar a cabo la detección y corrección de errores presentes en el mensaje transmitido, no obstante es de suma importancia tratar que la corrección de errores sea cercana al 100 % debido a que la aplicación es de uso médico, es por ello que deberá tener conocimiento completo del modelo de canal para lograr un buen resultado. Si desea ver más información sobre la corrección de errores puede consulta este [enlace](#).

Los requerimientos son dados como ruta de desarrollo es por ello que deberán ser respetados, no obstante la revisión será con base al funcionamiento de cada uno o también aspecto de investigación que se requiera. A continuación se detalla que requerimientos se esperan para este bloque:

1. El desarrollo de software deberá ser llevado a cabo en **Python** mediante el apoyo de **Jupyter Notebook**. **NOTA: No se aceptará el desarrollo de software en otro lenguaje, además se recomienda en el libro creado en Jupyter Notebook hacer comentarios de la implementación, para consulta de como comentar código ver este [enlace](#).**
2. El documento de Jupyter Notebook **deberá** estar almacenado en el GitHub respetando la estructura que el profesor le sugiera. **NOTA: Será revisado el aporte de los integrantes de cada grupo en desarrollo y commits que tenga el repositorio.**
3. La estrategia de como se implementará el proceso de decodificación de la información **será completamente abierta**, sin embargo el proceso de decodificación y manipulación de la información deberá ser explicado con detalle en la documentación (Informe final) además de comentarios en el archivo de **Jupyter**.
4. El sistema deberá ser capaz de hacer corrección de errores durante la ejecución del mismo, para ello deberá tomar en cuenta los parámetros de SNR o bien  $E_b/N_0$  utilizados en el canal. **NOTA: Por recomendación será importante que cada vez que se haga una ejecución se puede generar un archivo de texto plano que recopile información en relación a cantidad de errores según el parámetros de SNR o  $E_b/N_0$  utilizado para los enfoques de Hamming (7,4) y (15,11).**
5. Se deberá generar la tasa de error de bit o BER por cada ejecución completa, según los enfoques de Hamming utilizados (7,4 y (15,11), se recomienda que la información pueda ser almacenada en documento de texto plano esto facilita el manejo de información. **NOTA: En este [enlace](#) puede ver información relaciona al BER.**
6. En relación al desempeño de los enfoques de Hamming (7,4) y (15,11) determine la ganancia de código que existe y comente que desventajas y ventajas existen entre ambos enfoques, para ello puede consultar esta información en este [enlace](#).



7. Se deberá implementar en **Python** un método o función que permita medir el tiempo de ejecución transcurrido al ejecutar el script, para ello se recomienda revisar este [enlace](#) como método de referencia. **NOTA: Se deberá hacer al menos 200 ejecuciones del script para determinar un tiempo promedio de ejecución.**
8. Debido al estándar IEEE la tecnología BLE utiliza un sistema llamado CRC, no obstante al utilizar un sistema de redundancia tipo Hamming analice y argumente que ventajas y desventajas puede observar de usar CRC contra Hamming. Puede ver más información en detalle en el siguiente [enlace](#).
9. Finalmente el desarrollo de documentación será muy importante, es por ello que desde ya deberán ir estructurando su documento tipo paper formato **IEEE Transactions** a doble columna en idioma español. **NOTA: La documentación es muy importante debido a que tiene un puntaje por cada avance así que no será buena idea no hacerla, y procurar evitar errores gramaticales, ortografía y puntuación debido a que serán penalizados.**

### 5.3.3. Bloque Visualización

Este bloque deberá procesar la información recibida por el decodificador de tal manera que se puede organizar y se puedan mostrar gráficas o tablas, dicho bloque deberá ser implementado en Python para ello se recomienda utilizar funciones para manejo de datos tales como Matplotlib, Numpy y Pandas no genere scripts desde cero es preferible usar frameworks preestablecidos para graficación o manejo de datos.

Los requerimientos son dados como ruta de desarrollo es por ello que deberán ser respetados, no obstante la revisión será con base al funcionamiento de cada uno o también aspecto de investigación que se requiera. A continuación se detalla que requerimientos se esperan para este bloque:

1. El desarrollo de software deberá ser llevado a cabo en **Python** mediante el apoyo de **Jupyter Notebook**. **NOTA: No se aceptará el desarrollo de software en otro lenguaje, además se recomienda en el libro creado en Jupyter Notebook hacer comentarios de la implementación, para consulta de como comentar código ver este [enlace](#).**
2. El documento de Jupyter Notebook **deberá** estar almacenado en el GitHub respetando la estructura que el profesor le sugiera. **NOTA: Será revisado el aporte de los integrantes de cada grupo en desarrollo y commits que tenga el repositorio.**
3. El bloque de visualización deberá ser capaz de mostrar la información en al menos tres salidas, las cuales serán Zonas de entrenamiento, Gráficas de BER y Gráfico de Latidos por Minutos. **NOTA: Para este caso puede programar el script que haga las ejecuciones de manera automática y la información de visualización se pueda exportar en un archivo PDF final, esto se debe a la cantidad de ejecución y diversos escenarios que se deben implementar.**
4. Para la salida de Zonas de entrenamiento se deberá mostrar en una tabla los rangos temporales en donde se alcanzó la región calentamiento, quema grasa, aeróbica, anaeróbica y extrema, para ello utilice la información de latidos por minutos y edad mencionada en la Sección [5.1.1](#), si desea saber más información al respecto ver este [enlace](#).
5. Para la salida de grafica de BER se podrá utilizar un familia de curvas para reciclar gráficas, no obstante es importante trata de dejar los enfoque de Hamming (7,4) y (15,11) por aparte para evitar saturar los gráficos por cuestión de orden y legibilidad de la información.

6. Para la salida del gráfico de latidos por minuto se deberá confeccionar de tal manera que pueda recibir dos curvas en un mismo gráfico, debido a que el profesor en el Avance 3 le proporcionará los valores crudos de medición con el objetivo de determinar métricas de rendimiento tales como RMSE y Pbias donde responderán que tan eficaz fue el método de filtrado y la corrección de errores del mensaje. Si desea leer más al respecto de métricas de rendimiento para RMSE puede ver este [enlace](#) y para el Pbias este [enlace](#).
7. Se deberá implementar en **Python** un método o función que permita medir el tiempo de ejecución transcurrido al ejecutar el script, para ello se recomienda revisar este [enlace](#) como método de referencia. **NOTA: Se deberá hacer al menos 200 ejecuciones del script para determinar un tiempo promedio de ejecución.**
8. Finalmente el desarrollo de documentación será muy importante, es por ello que desde ya deberán ir estructurando su documento tipo paper formato **IEEE Transactions** a doble columna en idioma español, para ello pueden usar el template siguiente [acá](#). **NOTA: La documentación es muy importante debido a que tiene un puntaje por cada avance así que no será buena idea no hacerla, y procurar evitar errores gramaticales, ortografía y puntuación debido a que serán penalizados.**

## 6. Entregables y Evaluación

Esta sección tiene como objetivo de informar al estudiante como será la distribución de la evaluación a lo largo de la ejecución del proyecto. No obstante se adjunta la estructura del repositorio la cual se deberá generar para los entregables

### 6.1. Requerimientos para el Flujo de Trabajo

En relación con el repositorio de Git, se deben considerar varios aspectos. Es importante seguirlos para organizar el desarrollo de su código, documentación y preparar su repositorio:

1. Abra una cuenta de Git en un repositorio de Git gratuito (GitLab).
2. Cree un repositorio de Git con el siguiente nombre: `< grupo# > _TCE_2025`. Donde *group#* será el número de grupo asignado. Por ejemplo, para el Grupo 1, el nombre de su repositorio de Git será: `grupo1_TCE_2025`.
3. Por términos de evitar plagios el repositorio deberá ser privado y proporcione acceso a al usuario `sercr0388` (GitLab) y asegúrese de proporcionar permisos de escritura también.
4. El repositorio de Git debe contener dos ramas principales: `master` y `desarrollo`.
5. Inicialmente, la rama de desarrollo se crea a partir del `master`, tener en cuenta esto.
6. Cuando trabaje en un proyecto o desarrollo de prueba, el estudiante deberá crear una nueva rama de trabajo a partir de la rama `desarrollo` y cuando la función esté lista, la rama debe fusionarse para la rama de desarrollo. Cualquier corrección o modificación adicional después de la fusión debe requerir que se repita el proceso (es decir, crear la rama a partir de desarrollo y fusionar los cambios más tarde). Una vez que el código en la rama de desarrollo esté listo, se fusionará con



el maestro y se debe crear una etiqueta. El proceso se describe en la Figura 10. Se recomienda obtener más información [aquí](#).

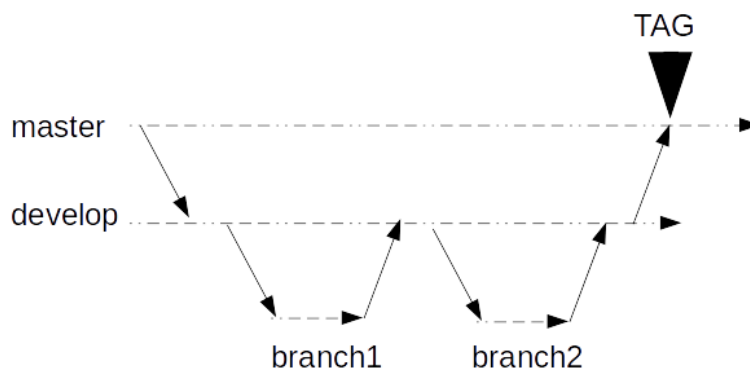


Figura 10: Estructura del repositorio para el proyecto.

## 6.2. Estructura del folder para los entregables

En relación con la documentación entregable y la estructura de carpetas, **debe** seguir la organización de la carpeta en su *usuario\_id.zip* y (*por ejemplo, grupo1\_TCE\_2023*) y el repositorio de Git como se muestra a continuación:

```

1  master ---- Proyecto_TCE
2          |-- Jupyter
3              |-- SistemaRF.ipynb
4          |-- Octave/Matlab
5              |-- ModDemod.m
6          |-- Eagle
7              |-- Board.sch
8              |-- Board.brd
9          |-- Documentacion
10             |-- Avance 1
11                 |-- Avance 1.pdf (LaTeX)
12             |-- Avance 2
13                 |-- Avance 2.pdf (LaTeX)
14             |-- Paper
15                 |-- Paper.pdf (LaTeX)
16             |-- Exposicion Final
17                 |-- Exposicion.pdf (LaTeX)
  
```

En relación a la estructura del repositorio, el objetivo es llevar el desarrollo del proyecto de manera organizada y de manera colaborativa. No obstante observe que sólo se están incluyen código fuente, archivos, resultados, etc. **NOTA: Por ninguna circunstancia agregue información con grandes tamaños debido a que GitHub permite una cantidad limitada de espacio.**

### 6.3. Evaluación

En esta sección se detallarán los puntos a evaluar durante el desarrollo del proyecto, es importante tomar en cuenta las recomendaciones realizadas en el instructivo esto le ayudará a no perder puntos en sus entregables a lo largo del desarrollo.

Para dar seguimiento del trabajo de cada grupo, se proyectan que se entregue al menos 2 avances durante el semestre lectivo sin embargo es importante acordar una cita grupal virtual **una semana antes de cada entregable** en términos de que el profesor le de realimentación. A continuación se dará detalle que contendrá cada avance además de la semana proyecta de entrega.

1. Avance 1: Este avance deberá ser entregado al final de la **semana 8** del semestre y contempla las siguientes actividades:
  - La unidad de transmisor **deberá** estar completa al 100 %, según los requerimientos planteados.
  - El modelo aproximado de canal deberá ser funcional al 50 %, según los requerimientos planteados.
  - La unidad de receptor deberá estar completa al 25 %, según los requerimientos planteados.
  - En relación al reporte **deberá** presentar un archivo en pdf el cual cumpla con el formato **IEEE Transactions** donde se presente una sección de Introducción, Metodología de Desarrollo, Análisis de Resultados (donde deberá hacer énfasis de los resultados obtenidos del TX), recuerde que toda información que utilice de terceros deberá ser referenciada (Trate de utilizar referencias de buen nivel académico y científico).
2. Avance 2: Este avance deberá ser entregado al final de la **semana 17** del semestre y contempla las siguientes actividades:
  - La unidad de receptor deberá estar completa al 100 %, según los requerimientos planteados.
  - El modelo aproximado de canal deberá ser funcional al 100 %, según los requerimientos planteados.
  - Bloque de visualización final completamente funcional al 85 %.
  - Comparación de eficiencia del sistema RF respecto a valores reales al 85 %.
  - En relación al reporte para el avance 1 este será acumulativo (entiéndase usar toda la información del avance 1 para hacer su avance 2) el mismo deberá presentar un archivo en pdf el cual cumpla con el formato **IEEE Transactions** donde se presente completa la Metodología incluyendo la técnicas de diseño de antena, receptor y además de validación de datos, no obstante es importante que en la sección de Análisis de Resultados deberá hacer discusión de los resultados obtenido por la antena y receptor. Finalmente recuerde que toda información que utilice de terceros deberá ser referenciada (Trate de utilizar referencias de buen nivel académico y científico).
3. Paper Final: El mismo ha sido generado a lo largo de cada avance entregado, es por ello que deberá ser entrega en semana 19 del semestre usando el formato **IEEE Transactions** y tendrá la siguiente estructura:
  - Título.

- Autores.
  - Resumen.
  - Introducción.
  - Metodología.
  - Análisis de Resultados.
  - Conclusiones
  - Recomendaciones.
  - Referencias.
4. Exposición: Deberá ser oral mediante un vídeo y comprende una exposición general del flujo de investigación y desarrollo que les permitieron dar solución al problema planteado, esta presentación está proyectada para que sea en semana 19 del semestre con un template Beamer  $\text{\LaTeX}$  y deberá tener la siguiente estructura:

- Portada.
- Introducción.
- Motivación.
- Problemática.
- Metodología de Desarrollo.
- Análisis de Resultados.
- Conclusiones.
- Recomendaciones.

Con base en la descripción de los entregables en la tabla 1 se muestra el detalle según los ítems descritos anteriormente. **NOTA: Para evitar confusiones es tomar en cuenta que el proyecto equivale a un 55 % de la evaluación del curso, sin embargo la calificación que se presentará en la tabla 1 representa como nota máxima de 100 % el cual representa el 55 % que el curso tiene destinado para el rubro de proyecto final.**

## 6.4. Entrega del Proyecto

Para la entrega del proyecto es necesario tomar en consideración lo siguiente:

1. Antes de enviar su proyecto revise que el formato y organización de archivos sea el que fue sugerido en la Sección 6.2.
2. Comprima el la carpeta entregable en un formato con .zip y entregue el mismo por correo electrónico y TEC-Digital, si tuviera problema indíquelo al profesor cuanto antes.
3. La fecha de entrega esta proyectada para el día 9 de Diciembre de 2025 hasta las 11:55 pm, superada la hora de entrega se toma tardío.

Tabla 1: Criterios de evaluación

Objeto de evaluación	Descripción	Porcentaje
<b>Avance 1</b>	La unidad de transmisor deberá estar completa al 100 %, según los requerimientos planteados. Modelo aproximado de canal deberá ser funcional al 50 %, según los requerimientos planteados. La unidad de receptor deberá estar completa al 25 %, según los requerimientos planteados. Cumple con el formato IEEE Transactions donde se presente una sección de Introducción, Metodología de Desarrollo, Análisis de Resultados (donde deberá hacer énfasis de los resultados obtenidos.)	<b>27 %</b>
<b>Avance 2</b>	La unidad de receptor deberá estar completa al 100 %. El modelo aproximado de canal deberá ser funcional al 100 % La antena seleccionada deberá estar implementada a nivel de PCB. Bloque de visualización final completamente funcional al 85 %. Comparación de eficiencia del sistema RF respecto a valores reales al 85 %. Deberá presentar un archivo en pdf el cual cumpla con el formato IEEE Transactions donde se presente una versión de al menos 90 % del trabajo.	<b>27 %</b>
<b>Paper Final</b>	Debe contener la información de los tres avances, usando un formato de IEEE y que posea la siguiente estructura: Título, Autores, Resumen, Introducción, Metodología, Análisis de Resultados, Conclusiones, Recomendaciones y Referencias.	<b>28 %</b>
<b>Exposición Final</b>	Cumple con la plantilla Beamer y contiene la estructura Portada, Introducción, Motivación, Problemática, Metodología de Desarrollo, Resultados, Conclusiones, Recomendaciones y Referencias.	<b>18 %</b>
<b>Total</b>		<b>100 %</b>

## Referencias

[1] Simon Haykin. *Communication Systems*. Wiley Publishing, 5th edition, 2009.