PROCESO GESTIÓN DEL TALENTO HUMANO

FORMATO INFORME MENSUAL EJECUCIÓN CONTRACTUAL

Barranquilla, Enero de 2022

Señor (a)
MANUEL GREGORIO HORMECHEA LANCE
SUPERVISOR(A) CONTRATO No. **3298199**
Coordinador Sede TIC
TIC Y Energía
Barranquilla

Asunto: Informe mensual de ejecución contractual
Mes Enero del año 2022

Referencia: No 3298199 del año 2022

LILIOLA PAOLA PRINS BALLESTAS, identificado con la cédula de ciudadanía No. 32.780.488 de Barranquilla, en mi calidad de Contratista del SENA, en TIC Y Energía, en cumplimiento del Contrato de Prestación de Servicios de la referencia, a continuación presento el Informe de actividades realizadas en el mes objeto de cobro.

Valor y forma de Pago: a). un pago correspondiente al mes de Enero de 2022 por valor de (811.491). Los honorarios serán pagados por el SENA al contratista de acuerdo al cronograma definido por la Dirección Administrativa y Financiera de la Dirección General, en la cuenta de Ahorros No. 620280107 BBVA, cuyo titular es el Contratista.

Plazo: Será hasta el 17de Diciembre de 2022.

OBJETO:
Prestar los servicios profesionales como Instructor para impartir formación en Programación de Software de acuerdo con la planeación de la coordinación de TIC y Energía del Centro Nacional Colombo Alemán

Obligaciones Especificas:

No	Obligaciones	Acciones realizadas	Evidencias
1	Impartir formación a las fichas asignadas	Seleccionar las estrategias de enseñanza, aprendizaje y evaluación pertinentes para el grupo. Orientar el proceso de aprendizaje de acuerdo con el diseño curricular vigente. Reportar la información académica y administrativa que se generen en el proceso.	Se realizó solicitud de creación de las fichas.
2	Impartir formación a las fichas asignadas	Sesiones de orientación mediadas por tecnología (Google Meeting - WhatsApp)	Capturas de pantalla tomadas de las sesiones

3	Impartir formación a las fichas asignadas	Espacios Virtuales disponibles en la plataforma institucional	Materiales y proyectos publicados en la plataforma

A continuación relaciono los desplazamientos que realicé previo a la presentación de este informe. Una vez finalizado cada desplazamiento presenté al ordenador del gasto el informe en el Formato Informe Legalización Desplazamiento Contratista GTH-F-087, en el que se describieron las actividades desarrolladas y los resultados de cada desplazamiento. Cada informe cuenta con el visto bueno del Supervisor.
Se lista a continuación el soporte de la legalización de los desplazamientos realizados, los cuales forman parte integral del presente informe de ejecución contractual.

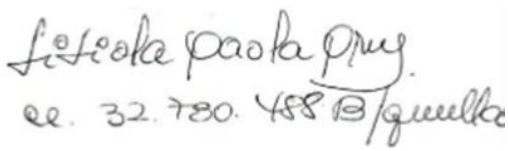
ITEM	No DE LA ORDEN DE VIAJE	LUGAR DE DESPLAZAMIENTO	FECHAS DE DESPLAZAMIENTOS	
			INICIO	FINALIZACIÓN

Nota 1: *Por cada desplazamiento que haya realizado el contratista, adjuntará el respectivo informe que la soporte. En caso de haber realizado el desplazamiento en fecha posterior a la presentación del informe de ejecución contractual, deberá reportarlo en el siguiente informe de ejecución contractual.*

Para el trámite de la cuenta me permito adjuntar: Documentos electrónicos enunciados como evidencias del cumplimiento de las obligaciones contractuales y los desplazamientos realizados y el No. 7820405740 de la planilla, operador y periodo. (Decreto Ley 2106 de 2019 – " *Decreto Ley Antitrámites"*)

Evidencias en () folios

Cordialmente,

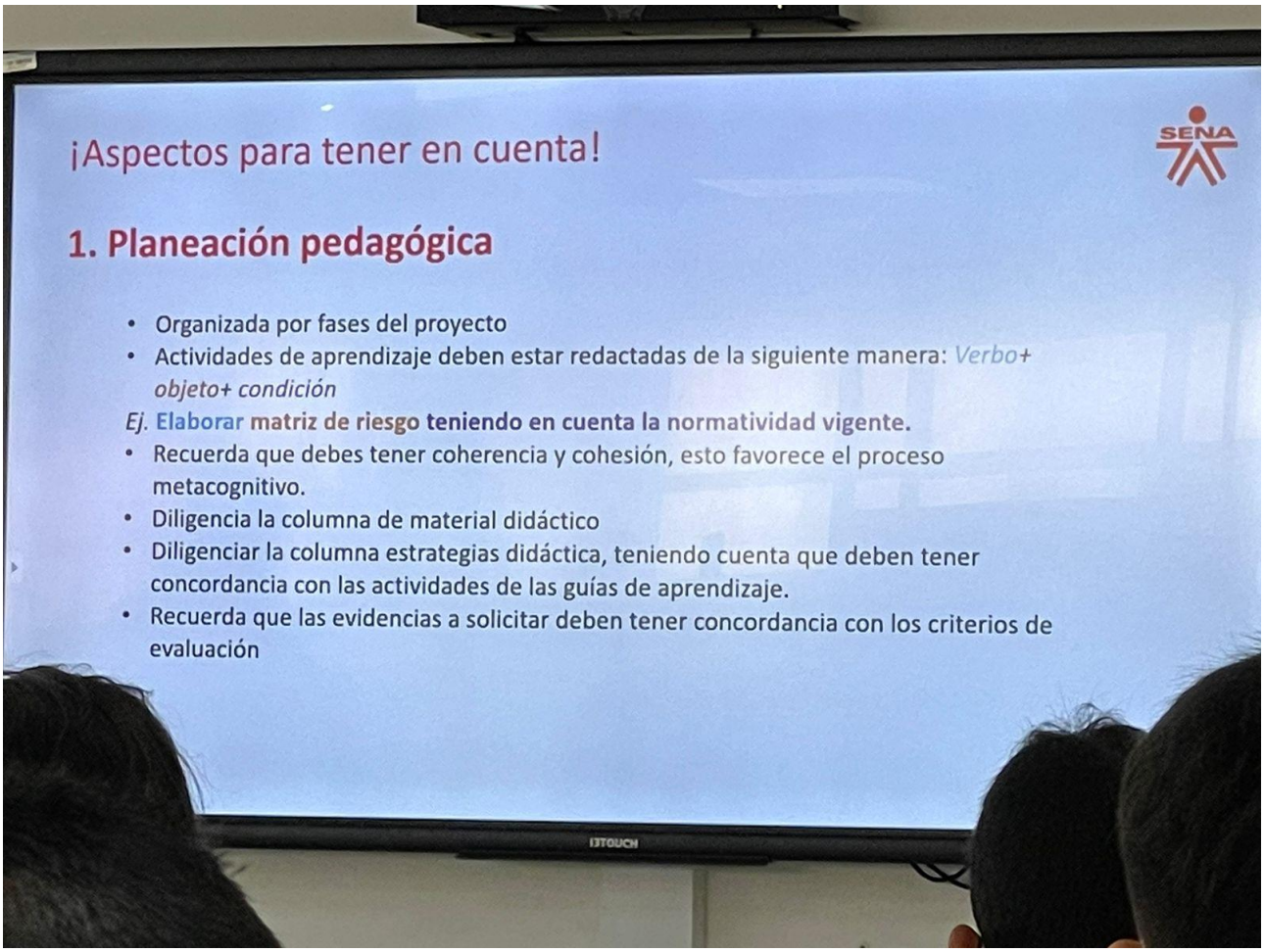


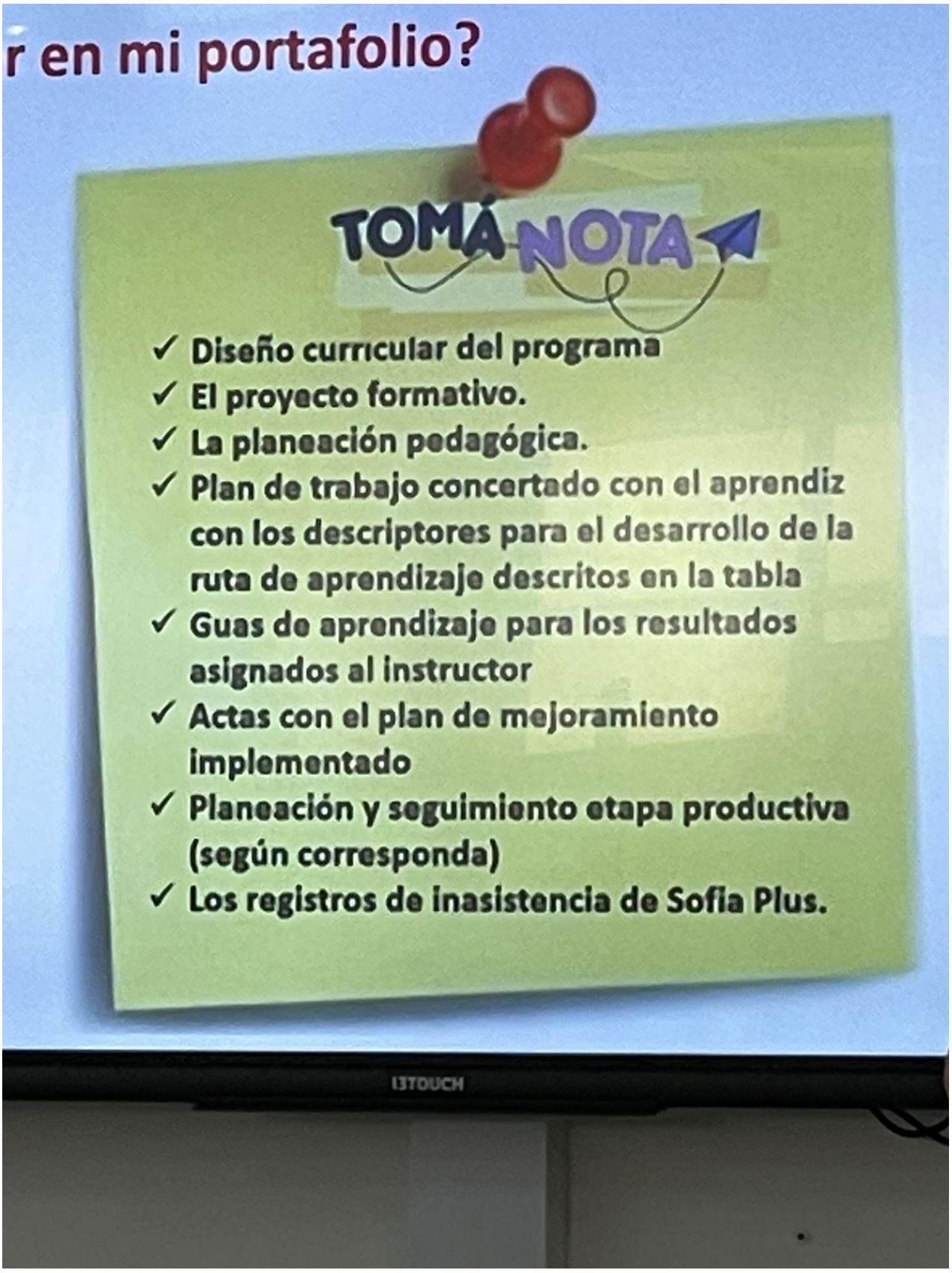
LILIOLA PAOLA PRINS BALLESTAS
Contratista
C.C. No. 32.780.488

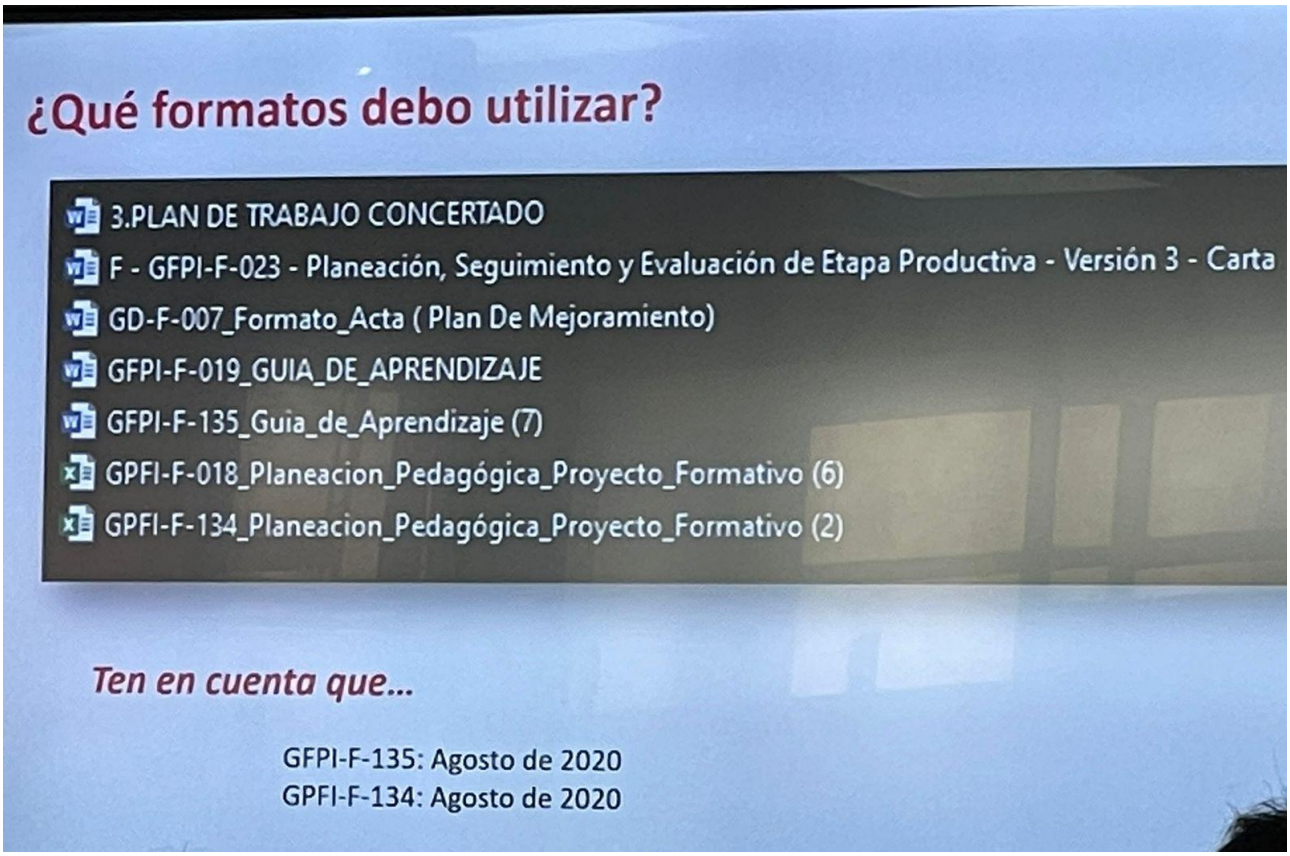
Recibí a satisfacción:

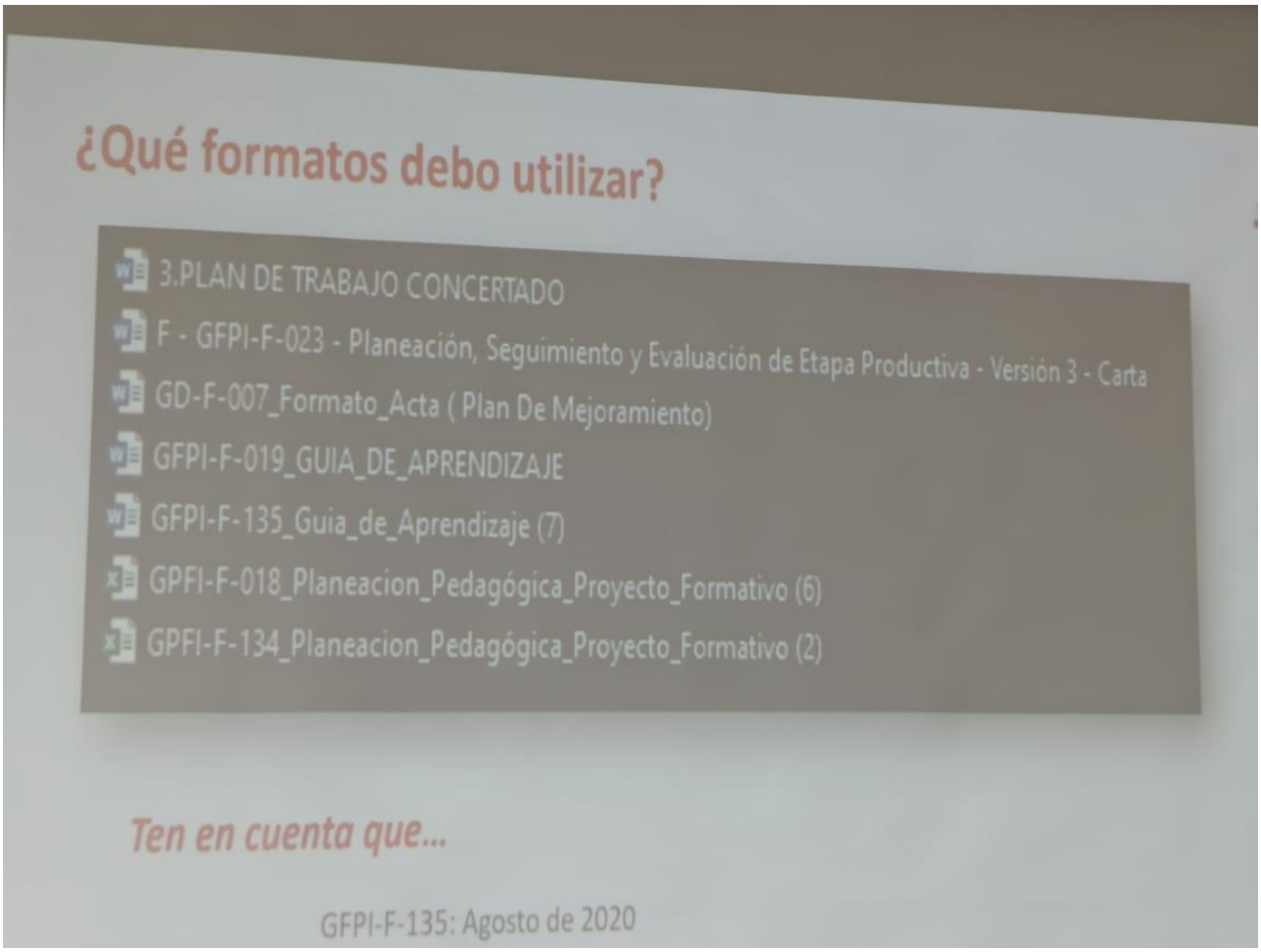
MANUEL GREGORIO HORMECHEA LANCE
Supervisor(a) Contrato 3298199 de 2022
Coordinador Sede TIC

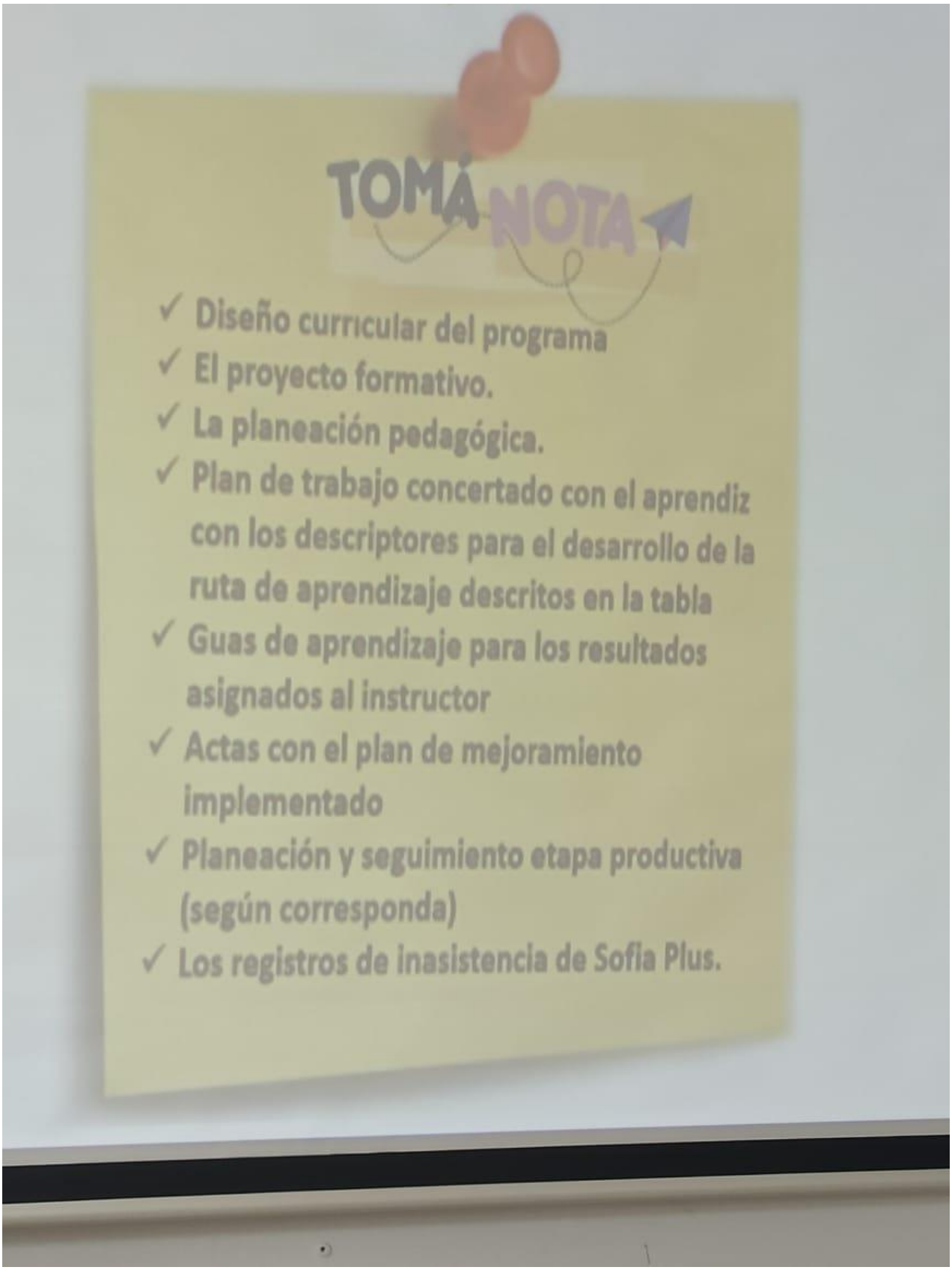
EVIDENCIAS


¡Aspectos para tener en cuenta!
1. Planeación pedagógica
• Organizada por fases del proyecto
• Actividades de aprendizaje deben estar redactadas de la siguiente manera: Verbo+ objeto+ condición
Ej. Elaborar matriz de riesgo teniendo en cuenta la normatividad vigente.
• Recuerda que debes tener coherencia y cohesión, esto favorece el proceso metacognitivo.
• Diligencia la columna de material didáctico
• Diligenciar la columna estrategias didáctica, teniendo cuenta que deben tener concordancia con las actividades de las guías de aprendizaje.
• Recuerda que las evidencias a solicitar deben tener concordancia con los criterios de evaluación


r en mi portafolio?
TOMÁ NOTA
✓ Diseño curricular del programa
✓ El proyecto formativo.
✓ La planeación pedagógica.
✓ Plan de trabajo concertado con el aprendiz con los descriptores para el desarrollo de la ruta de aprendizaje descritos en la tabla
✓ Guas de aprendizaje para los resultados asignados al instructor
✓ Actas con el plan de mejoramiento implementado
✓ Planeación y seguimiento etapa productiva (según corresponda)
✓ Los registros de inasistencia de Sofia Plus.
I3TOUCH


¿Qué formatos debo utilizar?
3.PLAN DE TRABAJO CONCERTADO
F - GFPI-F-023 - Planeación, Seguimiento y Evaluación de Etapa Productiva - Versión 3 - Carta
GD-F-007_Formato_Acta (Plan De Mejoramiento)
GFPI-F-019_GUIA_DE_APRENDIZAJE
GFPI-F-135_Guia_de_Aprendizaje (7)
GPFI-F-018_Planeacion_Pedagógica_Proyecto_Formativo (6)
GPFI-F-134_Planeacion_Pedagógica_Proyecto_Formativo (2)
Ten en cuenta que...
GFPI-F-135: Agosto de 2020
GPFI-F-134: Agosto de 2020


¿Qué formatos debo utilizar?
3.PLAN DE TRABAJO CONCERTADO
F - GFPI-F-023 - Planeación, Seguimiento y Evaluación de Etapa Productiva - Versión 3 - Carta
GD-F-007_Formato_Acta (Plan De Mejoramiento)
GFPI-F-019_GUIA_DE_APRENDIZAJE
GFPI-F-135_Guia_de_Aprendizaje (7)
GPFI-F-018_Planeacion_Pedagógica_Proyecto_Formativo (6)
GPFI-F-134_Planeacion_Pedagógica_Proyecto_Formativo (2)
Ten en cuenta que...
GFPI-F-135: Agosto de 2020


TOMA NOTA
✓ Diseño curricular del programa
✓ El proyecto formativo.
✓ La planeación pedagógica.
✓ Plan de trabajo concertado con el aprendiz con los descriptores para el desarrollo de la ruta de aprendizaje descritos en la tabla
✓ Guas de aprendizaje para los resultados asignados al instructor
✓ Actas con el plan de mejoramiento implementado
✓ Planeación y seguimiento etapa productiva (según corresponda)
✓ Los registros de inasistencia de Sofia Plus.

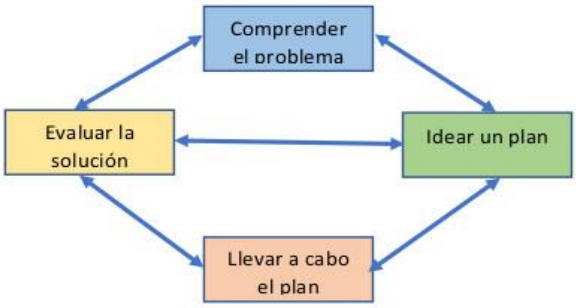

Claro
Claro
Claro
Claro
Claro
Claro
Claro
Claro


RUTA DE EVACUACIÓN
movistar

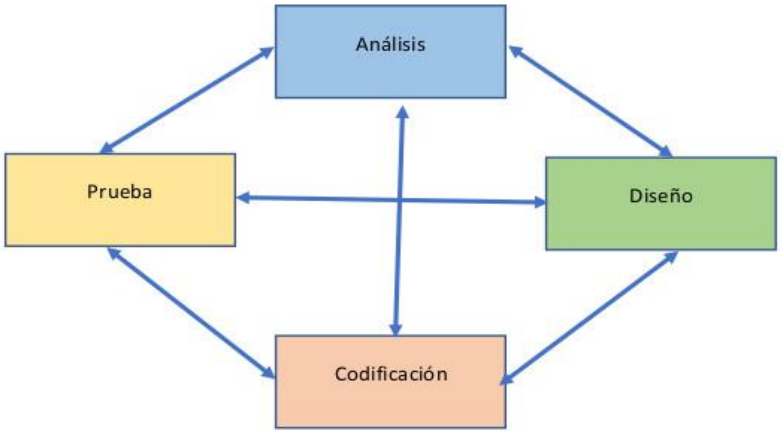
METODOLOGIA HOLÍSTICA

PROPUESTA

La metodología Holística , integra el método descriptivo de solución de problemas, el cual comprende cuatro fases perfectamente diferenciables, aun cuando no necesariamente disjuntas, dichas fases son: 1) **Comprender el problema**, 2) **Idear un plan** (Formular una estrategia general – proceso inductivo -) 3) **Llevar a cabo el plan** (Probar la estrategia formulada en el punto anterior – proceso deductivo -) y 4) **Evaluar la solución** (Verificar los resultados). Dichas etapas y su interrelación se esquematizan en la gráfica que sigue a continuación.



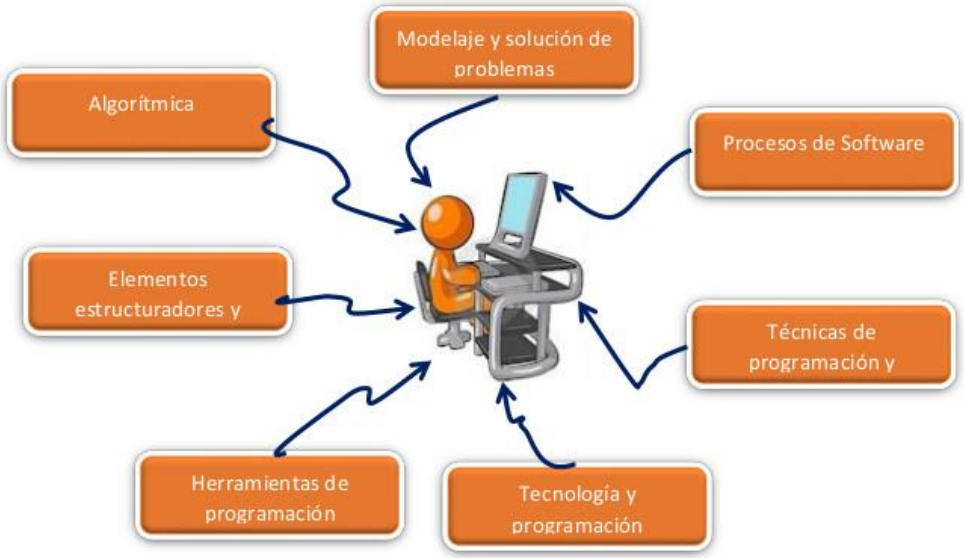
En términos de desarrollo de software podemos representar esta grafica de la siguiente forma;



Ingeniera Liliola Paola Prins Ballestas- ocdt2020

METODOLOGIA HOLÍSTICA

El desarrollo sistémico seria desarrollado en las siguientes etapas



Modelado y solución de problemas: Es la capacidad de abstraer la información de la realidad relevante para un problema, de expresar dicha realidad en términos de algún lenguaje y proponer una solución en términos de modificaciones de dicha abstracción. Se denomina "análisis" al proceso de crear dicha abstracción a partir de la realidad, y "especificación del problema" al resultado de expresar el problema en términos de dicha abstracción.

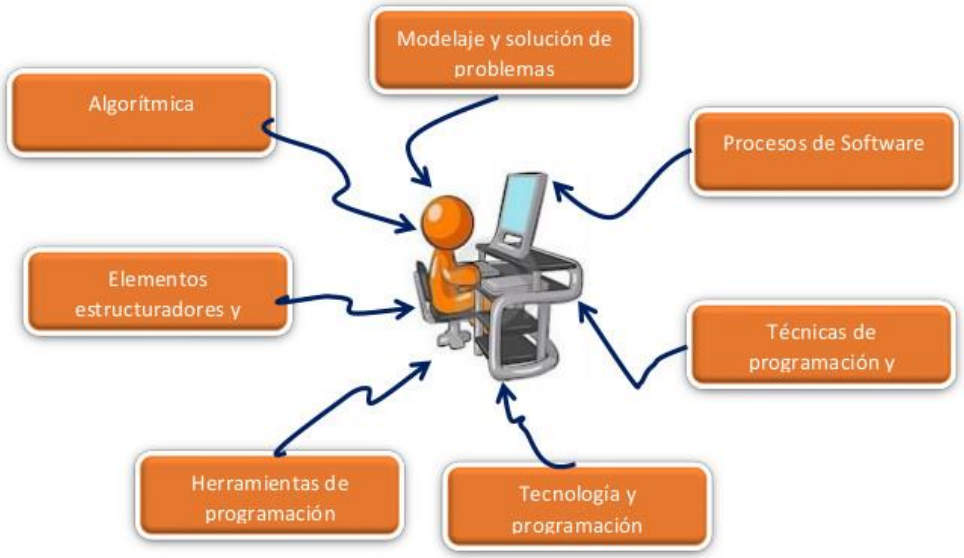
Algorítmica: Es la capacidad de utilizar un conjunto de instrucciones para expresar las modificaciones que se deben hacer sobre la abstracción de la realidad, para llegar a un punto en el cual el problema se considere resuelto. Se denomina "diseño de un algoritmo" al proceso de construcción de dicho conjunto de instrucciones.

Tecnología y programación: Son los elementos tecnológicos necesarios (lenguaje de programación, lenguaje de modelado, etc.) para expresar, en un lenguaje comprensible por una máquina, la abstracción de la realidad y el algoritmo que resuelve un problema sobre dicha abstracción. Programar es la habilidad de utilizar dicha tecnología para que una máquina sea capaz de resolver el problema.

Ingeniera Liliola Paola Prins Ballestas- ocdt2020

METODOLOGIA HOLÍSTICA

El desarrollo sistémico seria desarrollado en las siguientes etapas



Modelado y solución de problemas: Es la capacidad de abstraer la información de la realidad relevante para un problema, de expresar dicha realidad en términos de algún lenguaje y proponer una solución en términos de modificaciones de dicha abstracción. Se denomina "análisis" al proceso de crear dicha abstracción a partir de la realidad, y "especificación del problema" al resultado de expresar el problema en términos de dicha abstracción.

Algorítmica: Es la capacidad de utilizar un conjunto de instrucciones para expresar las modificaciones que se deben hacer sobre la abstracción de la realidad, para llegar a un punto en el cual el problema se considere resuelto. Se denomina "diseño de un algoritmo" al proceso de construcción de dicho conjunto de instrucciones.

Tecnología y programación: Son los elementos tecnológicos necesarios (lenguaje de programación, lenguaje de modelado, etc.) para expresar, en un lenguaje comprensible por una máquina, la abstracción de la realidad y el algoritmo que resuelve un problema sobre dicha abstracción. Programar es la habilidad de utilizar dicha tecnología para que una máquina sea capaz de resolver el problema.

Ingeniera Liliola Paola Prins Ballestas- ocdt2020



```
'''
Este programa representa el juego de adivinar un número en n intentos; para lo cual se:
   a) Selecciona aleatoriamente (piensa) un número de 4 cifras.
   b) Se permite un máximo de 15 intentos
   c) Por cada uno de los intentos, el jugador indica un número y:
      1) Si el número "pensado" por el computador es mayor se indica mayor
      2) Si el número "pensado" por el computador es menor se indica menor
      3) Si los números coinciden se indica que ganó.

ANALISIS
   META       :  Diseñar el programa que permita jugar

   ENTRADAS   :  num    :  número ingresado por el jugador para intentar adivinar

   SALIDAS    :  msg    :  mensaje indicando si el jugador logró o no adivinar el
                           número

   VARIABLES  :  num_pen :  Número seleccionado (pensado) por el computador
                 num_int :  Número máximo de intentos para adivinar el número
                 intento :  En que intento va el jugador

   PROCESO    :  a)  El computador (programa) genera un número
                 b)  Se inicializan todas las variables de proceso
                 c)  En un ciclo repetitivo que se ejecuta hasta que se adivine el
                     número o hasta que se agoten los intentos
                     0)  Se incrementa el número de intento
                     1)  El jugador indica un número
```

```
            2)  El programa informa mas alto, mas bajo o ganaste
ALGORITMO
   Inicio
      num_pen <- aleatorioentre(1000,9999)
      num_int <- 15
      intento <- 1
      num <- 0
      MQ intento<num_int y num<>num_pen Haga
         lea num
         Si num<>num_pen Entonces
            Si num<num_pen Entonces
               Escriba "mas alto"
            Sino
               Escriba "mas bajo"
            FinSi
            intento <- intento + 1
         FinSi
      FMQ
      Si num=num_pen Entonces
         Escriba "Felicitaciones, lo lograste en ",intento," intentos"
      Sino
         Escriba "Lo siento, se agotaron los turnos. El número a adivinar era ",num_pen
      FinSi
   Fin

COMENTARIOS
   La versión implementada en pyton, introduce validación de entradas para evitar
```

fallos del programa al momento que el usuario introduce un valor.

RESULTADOS DE UN CASO DE PRUEBA.

Intento 1 : 1

Por favor indique un número entero válido entre 1000 y 9999

Intento 1 : 5000

Mas alto

Intento 2 : 7500

Mas alto

Intento 3 : 8700

Mas bajo

Intento 4 : 8000

Mas alto

Intento 5 : 8350

Mas bajo

Intento 6 : 8200

Mas bajo

Intento 7 : 8100

Mas alto

Intento 8 : 8150

Mas bajo

Intento 9 : 8125

Mas alto

Intento 10 : 8140

Mas alto

Intento 11 : 8145

Mas bajo

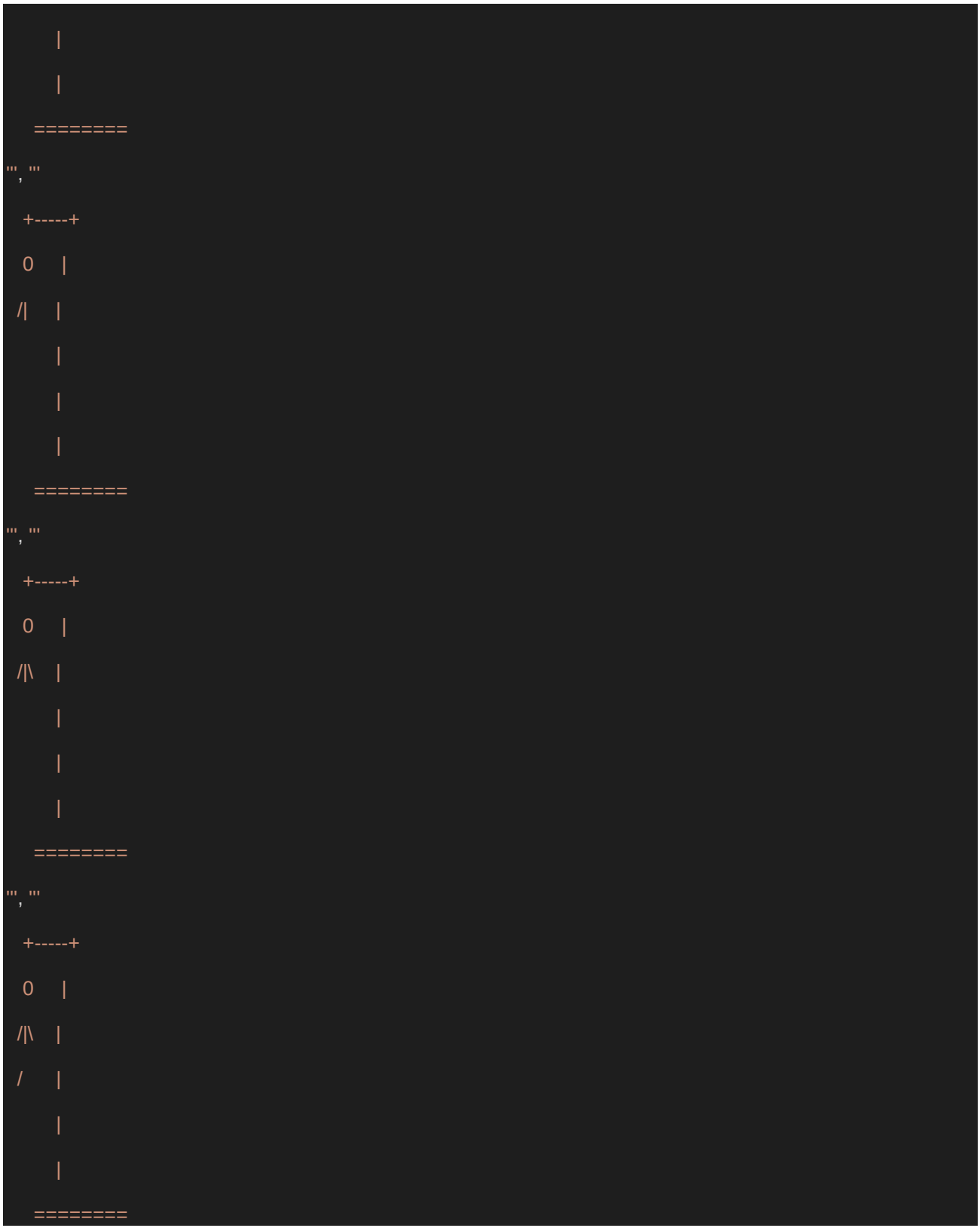
```
    Intento 12 : 8142
    Mas alto
    Intento 13 : 8144
    Mas bajo
    Intento 14 : 8143
    Felicidades lo lograste en 14 intentos
'''
from random import randint    ## Se selecciona el módulo para generar (pensar) el número
def entradaentera(msg='Digite un número entero', men=0, may=9999999999):
    while True:
        try:
            num = int(input(msg))
            if men<=num<=may:
                break
            else:
                print(f'Por favor indique un número entero válido entre {men} y {may}')
        except Exception:
            print(f'Por favor indique un número entero válido entre {men} y {may}')
    return num

num_pen = randint(1000,9999)  ## Se genera (piensa), el número
num_int = 15                  ## Se establece el número máximo de intentos
intento = numero = 0          ## No se ha realizado el primer intento
while intento<num_int and numero!=num_pen:
    intento += 1
    numero = entradaentera(f'Intento {intento:2d} : ',1000, 9999)
    if numero<num_pen:
```

```
        print("Mas alto")
    elif numero>num_pen:
        print("Mas bajo")
if numero==num_pen:
    print(f'Felicidades lo lograste en {intento} intentos')
else:
    print(f'Sigue intentando, el número que pensé fue :: {num_pen}')


##
## NOTA: Aplicar las modificaciones necesarias para que el programa permita jugar mas de
# una vez
##
```

```
import os
from random import randint
palabra   = ""
correctas = []
errores   = []
rpta      = 'si'
partes    = ['''
  +-----+
        |
        |
        |
        |
        |
    ========
''', '''
  +-----+
  0     |
        |
        |
        |
        |
    ========
''', '''
  +-----+
  0     |
  |     |
        |
```


|
|
========
'''
+-----+
0 |
/| |
|
|
|
========
'''
+-----+
0 |
/|\ |
|
|
|
========
'''
+-----+
0 |
/|\ |
/ |
|
|
========

```
''', '''
  +-----+
  0     |
 /|\    |
 / \    |
        |
        |
    ========
''']
palabras  = ['software', 'windows', 'computador', 'análisis', 'diseño', 'trabajo', 'programa', 'aplicativo', 'app',
'prueba', 'documentación', 'scrum', 'cascada', 'waterfall', 'algoritmo', 'lógica']

def nuevapalabra(lista) -> str:
    """ Selecciona de forma aleatoria una palabra de la lista """
    try:
        palabra = lista[randint(0, len(lista)-1)]
    except Exception:
        palabra = 'bailarina'
    return palabra

def cambio(letra) -> str:
    """ Ignora las tildes, retornando la letra sin tilde """
    try:
        if letra =='á':
            letra = 'a'
        elif letra =='é':
            letra = 'e'
```

```
        elif letra=='í':
            letra = 'i'
        elif letra=='ó':
            letra = 'o'
        elif letra in 'úü':
            letra = 'u'
    except Exception:
        pass
    return letra


def mostrar(palabra, correctas, errores) -> None:
    """ Despliega el estado del juego """
    try:
        print("AHORCADO")
        print(partes[len(errores)])
        res = 'Intentos fallidos: '
        for letra in errores:
            res = res+f'{letra} '
        print(res)
        enblanco = ""
        for letra in palabra:
            if cambio(letra) in correctas:
                enblanco = enblanco+letra
            else:
                enblanco = enblanco+"_"
        print(enblanco)
    except Exception:
```

```
        pass


def leer(entradas) -> str:
    """ Toma una letra de la entrada y realiza las validaciones necesarias """
    sw = True
    try:
        while sw:
            entrada = input("Ingrese una letra: ").strip().lower()
            if len(entrada) != 1:
                print("Por favor escriba una sola letra.")
            elif entrada in entradas:
                print("Letra repetida, ya fue utilizada.")
            elif entrada not in "abcdefghijklmnñopqrstuvwxyz":
                print("Debe indicar una letra.")
            else:
                sw = False
    except Exception:
        pass
    return entrada


def cls() -> None:
    """ Se encarga de borrar la pantalla dependiendo del sistema operativo """
    if os.name == 'nt':
        os.system('cls')
    else:
        os.system('clear')
```

```
def valida(letra,palabra) -> bool:
    """ Se encarga de verificar si una letra se encuentra en la palabra, ignora las tildes """
    try:
        sw = False
        if letra=='a' and ('a' in palabra or 'á' in palabra):
            sw = True
        elif letra=='e' and ('e' in palabra or 'é' in palabra):
            sw = True
        elif letra=='i' and ('i' in palabra or 'í' in palabra):
            sw = True
        elif letra=='o' and ('o' in palabra or 'ó' in palabra):
            sw = True
        elif letra=='u' and ('u' in palabra or 'ú' in palabra or 'ü' in palabra):
            sw = True
        else:
            sw = letra in palabra
    except Exception:
        pass
    return sw


def iniciar() -> None:
    """ Inicializa las variables del juego """
    global palabra
    global correctas
    global errores
    global rpta
    palabra = nuevapalabra(palabras)
```

```
    correctas = []
    errores = []
    rpta = 'si'
    cls()
    mostrar(palabra,correctas,errores)


iniciar()
while rpta.startswith('s'):
    letraEntrada = leer(errores+correctas)
    if valida(letraEntrada,palabra):
        correctas.append(letraEntrada)
    else:
        errores.append(letraEntrada)
    cls()
    mostrar(palabra, correctas, errores)
    todas = True
    for letra in palabra:
        if cambio(letra) not in correctas:
            todas = False
            break
    if todas or len(errores) == len(partes)-1:
        if todas:
            print("Excelente, GANASTE.")
        else:
            print("Lo siento, PERDISTE.")
        print(f"Intentos perdidos  : {len(errores)}")
        print(f"Intentos correctos : {len(correctas)}")
```

```
print(f"La palabra era      : {palabra}")
rpta = input("Desea jugar otra partida [Si/No] : ").lower()
if rpta.startswith('s'):
    iniciar()
```