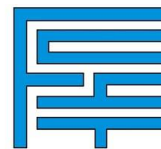




UNIVERSIDAD MAYOR DE SAN SIMÓN
DIPLOMADO CIENCIA DE DATOS



Proyecto Final 2

Cadena de suministro

Integrante

Anahi Gomez Nina

Docente

Msc. Ariel Zeballos

PRIMERA PARTE

A. IBP

BALANCED SCORECARD				
Empresa de comida rápida: Pollos Don Lucho				
VISION	Ser la mejor empresa de comida rápida en el mercado local, ofreciendo comida de calidad y atención eficiente a los clientes y consumidores, con personal calificado, utilizando tecnología actualizada.			
MISION	Satisfacer las necesidades de alimentación de los clientes y consumidores, ofreciendo la mejor comida de calidad para disfrutar en familia, lograr un máximo crecimiento con la continua innovación.			
PRIORIDAD ESTRATEGICA	Aumentar los clientes	Calidad del producto		
RESULTADOS DE LA ESTRATEGIA	Las utilidades serán mayores	Fidelidad de clientes		
PERPECTIVA	OBJETIVOS ESTRATEGICOS	INDICADORES	METAS	INICIATIVAS
FIANANCIERO (Mide y obtiene un buen conocimiento de la situación de la empresa)	Incrementar las ventas.	Margen de utilidad sobre ventas.	↑ 15% mensual	Definir políticas estándar de Precio.
	Aumentar utilidades.	% de ventas mensuales	↑ 20% sobre las ventas trimestre	Análisis de estados financieros.
		Margen de contribución mensual		
	Garantizar sostenibilidad del crecimiento	% cantidad de clientes nuevos	10 clientes nuevos/mes	Incremento en gastos de publicidad.
		Índice de Prueba Acida		
CLIENTE (Determina la situación de la empresa en el mercado.)	Conservar fidelidad de clientes actuales.	% clientes fieles	0 % deserción de clientes./trimestre	Acumular puntos por consumo.
	Conseguir popularidad en el mercado	% de participación en el mercado	70 % de clientes captados./trimestre	Programas de publicidad y Promociones.
	Garantizar que todo cliente cumpla sus expectativas	% de clientes satisfechos	90% clientes satisfechos/ trimestre	Control de calidad del producto.
	Disminución de clientes insatisfechos	Índice de quejas y reclamos	8% de quejas y reclamos./mes	Buzón de quejas y reclamos
				Encuestas.

PROCESOS INTERNOS (Modo de trabajo para conseguir los objetivos)	Optimizar procesos de producción.	% de verificación de optimización positiva	80% de la verificación/trimestre	Rediseño de procesos de producción.
	Mejorar atención al cliente	Índice de tiempo de preparación de un pedido	Mínimo 18 min. tiempo de espera	Actividades de distracción para clientes
	Mejorar el sistema de información.	% de clientes informados	60% de clientes informados/trimestre	Redes sociales, publicidad, reservas online
	Disminuir costo de producción.	% de reducción del costo de producción	Reducir 10% costo de producción/trimestre	Crear sistema de inventario
				Control de pérdidas proceso de producción.
ORGANIZATIVO/ CAPACIDAD (modo de actuar para seguir la estrategia)	Estimular la motivación.	% personal motivado.	90% personal motivado	Incentivos y actividades de recreación para personal.
	Infraestructura adecuada para el personal	% personal satisfecho con la infraestructura	90% personal satisfecho./trimestre	Reestructuración estratégica del inmueble.
	Capacitación del personal	% personal capacitado	100% personal capacitado/semestre	Cursos de: atención al cliente, Seguridad

B. LINEAR AND MIP(Librería pulp en Python)

Es una librería de optimización.

Fuente: <https://whdeveloper.wordpress.com/2019/04/18/desarrollo-de-una-aplicacion-en-python-para-resolucion-de-problemas-de-programacion-lineal-caso-transportes-parte-2/>

```
[1] #Importamos las librerías necesarias para realizar la optimización
import pandas as pd
import numpy as np
!pip install pulp
import pulp

# https://es.switch-case.com/45729413
```

Collecting pulp
Downloading <https://files.pythonhosted.org/packages/fb/34/ff5915ff6bae91c7c4cc22c3c369a6aea0b2127045dd5f308a91c260ac/PuLP-2.0-py3-none-any.whl>
39.2MB 104kB/s
Requirement already satisfied: pyparsing>=2.0.1 in /usr/local/lib/python3.6/dist-packages (from pulp) (2.4.5)
Installing collected packages: pulp
Successfully installed pulp-2.0

```
from pulp import * # importamos la librería pulp
# Definimos estructura de los datos para nuestro problema
# Definiendo los productos
products = ['cola cola', 'mani', 'queso', 'cerveza']
itemsets = ['x1', 'x2', 'x3']
#Costos de los productos
costs = {'cola cola': 5, 'mani': 3, 'queso': 1, 'cerveza': 4 }
# Definimos agrupaciones más frecuentes (Combos)
itemset_dict = {'x1': (('cola cola', 'mani'),10),
                'x2': (('mani', 'queso'),20),
                'x3': (('mani', 'cerveza'),30)}
#Definimos el problema para maximizar
my_lp_program = LpProblem('My_LP_Problem', LpMaximize)
# Definimos variables de tipo binario
products_var=LpVariable.dicts("Products", products, cat='Binary')
itemsets_var=LpVariable.dicts("Itemsets", itemsets, cat='Binary')
#Definimos los objetivos
my_lp_program += LpAffineExpression([(itemsets_var[x], itemset_dict[x][1]) for x in itemsets_var]) + \
    LpAffineExpression([(products_var[x], -costs[x]) for x in products_var])

# Definimos restricciones
my_lp_program += lpSum(products_var) <= 3, 'Constraint'
```

```
counter = 1
for a in itemset_dict.keys():
    item = itemsets_var[a]
    for b in itemset_dict[a][0]:
        product = products_var[b]
        counter +=1
        my_lp_program += product >= item, "{}Constraint".format(counter)
my_lp_program.writeLP("CheckLpProgram.lp")
my_lp_program.solve()
print("Estado:", LpStatus[my_lp_program.status])
print("Total Optimo=", value(my_lp_program.objective))
for v in my_lp_program.variables():
    print(v.name, "=", v.varValue)
```

Estado: Optimal
Total Optimo= 42.0
Itemsets_x1 = 0.0
Itemsets_x2 = 1.0
Itemsets_x3 = 1.0
Products_cerveza = 1.0
Products_cola_cola = 0.0
Products_mani = 1.0
Products_queso = 1.0

C. EVENTOS DE SIMULACIÓN DISCRETA (Librería simpy en Python)

Simulación de cola de espera en el Banco

Fuente: https://simpy.readthedocs.io/en/latest/examples/bank_renege.html

<https://pythonhosted.org/SimPy/Tutorials/TheBank.html>

```
[1] import math
    !pip install simpy
    import simpy
```

```
Collecting simpy
  Downloading https://files.pythonhosted.org/packages/5a/64/8f0fc71400d41b6c2c6443d333a1cade458d23d4945ccda700c810f
Installing collected packages: simpy
Successfully installed simpy-3.0.11
```

```
[2] # Atención en el Banco
    SEMILLA = 20
    NUM_CAJEROS = 3
    TIEMPO_ATENCION_MIN = 15
    TIEMPO_ATENCION_MAX = 35
    T_LLEGADAS = 10
    TIEMPO_SIMULACION = 90
    TOT_CLIENTES = 5

    te = 0.0 # tiempo de espera total
    dt = 0.0 # duracion de servicio total
    fin = 0.0 # minuto en el que finaliza
```

```
# Funcion de atencion al cliente
def Atencion(cliente):
    global dt #Accedemos a la variable dt declarada anteriormente
    # numero aleatorio y lo guarda en R
    R = random.random()
    tiempo = TIEMPO_ATENCION_MAX - TIEMPO_ATENCION_MIN
    # Distribucion uniforme
    tiempo_aten = TIEMPO_ATENCION_MIN + (tiempo*R)
    # deja correr el tiempo n minutos
    yield env.timeout(tiempo_aten)
    print("3er paso: Atencion listo a %s en %.2f minutos" % (cliente, tiempo_aten))
    # Acumula los tiempos de uso de la i
    dt = dt + tiempo_aten

#
def cliente (env, nombre, personal):
    global te
    global fin
    # Guarda el minuto de llegada del cliente
    llega = env.now
    print ("1ro paso: %s llego al Banco en el minuto %.2f" % (nombre, llega))
    # Espera su turno
    with personal.request() as request:
        # Obtiene turno
        yield request
```

```

# Guarda el minuto cuando comienza a ser atendido
pasa = env.now
# Calcula el tiempo que espero
espera = pasa - llega
# Acumula los tiempos de espera
te = te + espera
print ("2do paso: %s pasa con el cajero en el minuto %.2f habiendo esperado %.2f" % (nombre, pasa, espera))
# Invoca al proceso Atencion
yield env.process(Atencion(nombre))
#Guarda el minuto en que termina el proceso Atencion
deja = env.now
print ("4to paso: %s deja el Banco en el minuto %.2f" % (nombre, deja))
# Conserva globalmente el ultimo minuto de la simulacion
fin = deja

def principal (env, personal):
    llegada = 0
    i = 0
    # Para 10 clientes
    for i in range(TOT_CLIENTES):
        R = random.random()
        # Distribucion exponencial
        llegada = -T_LLEGADAS * math.log(R)
        # Deja transcurrir un tiempo entre uno y otro
        yield env.timeout(llegada)
        i += 1
        env.process(cliente(env, 'Cliente %d' % i, personal))

print ("===== SIMULACION COLA EN EL BANCO=====")
# Cualquier valor
random.seed (SEMILLA)
# Crea el objeto entorno de simulacion
env = simpy.Environment()
#Crea los recursos
personal = simpy.Resource(env, NUM_CAJEROS)
#Invoca el proceso prncial
env.process(principal(env, personal))
#Inicia la simulacion
env.run()

```

```

===== SIMULACION COLA EN EL BANCO=====
1ro paso: Cliente 1 lleo al Banco en el minuto 0.99
2do paso: Cliente 1 pasa con el cajero en el minuto 0.99 habiendo esperado 0.00
1ro paso: Cliente 2 lleo al Banco en el minuto 4.76
2do paso: Cliente 2 pasa con el cajero en el minuto 4.76 habiendo esperado 0.00
1ro paso: Cliente 3 lleo al Banco en el minuto 5.76

```

```

1ro paso: Cliente 4 llego al Banco en el minuto 10.29
1ro paso: Cliente 5 llego al Banco en el minuto 11.66
3er paso: Atencion listo a Cliente 2 en 20.20 minutos
4to paso: Cliente 2 deja el Banco en el minuto 24.95
2do paso: Cliente 4 pasa con el cajero en el minuto 24.95 habiendo esperado 14.66
3er paso: Atencion listo a Cliente 1 en 30.33 minutos
4to paso: Cliente 1 deja el Banco en el minuto 31.32
2do paso: Cliente 5 pasa con el cajero en el minuto 31.32 habiendo esperado 19.66
3er paso: Atencion listo a Cliente 3 en 33.10 minutos
4to paso: Cliente 3 deja el Banco en el minuto 38.86
3er paso: Atencion listo a Cliente 5 en 18.39 minutos
4to paso: Cliente 5 deja el Banco en el minuto 49.71
3er paso: Atencion listo a Cliente 4 en 26.46 minutos
4to paso: Cliente 4 deja el Banco en el minuto 51.41

```

```

[4] print ("\n-----")
    print ("\nINDICADORES: ")
    lpc = te / fin
    print ("\nLongitud promedio de la cola: %.2f" % lpc)
    tep = te / TOT_CLIENTES
    print ("Tiempo de espera promedio = %.2f" % tep)
    upi = (dt / fin) / NUM_CAJEROS
    print ("Uso promedio de la instalacion = %.2f" % upi)
    print ("\n-----")

```



```

-----

INDICADORES:

Longitud promedio de la cola: 0.67
Tiempo de espera promedio = 6.87
Uso promedio de la instalacion = 0.83

```

D. CONTROL DE PROCESOS (Librería pyspc en Python)

Producción de dulces en los Estados Unidos por fecha gestión y mes.

Fuente: https://www.kaggle.com/search?q=candy_production

```
[1] import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
import warnings
!pip install pyspc
from pyspc import *
warnings.filterwarnings('ignore') # Para evitar los molestos avisos.
%matplotlib inline
```

Collecting pyspc

Downloading <https://files.pythonhosted.org/packages/f1/d6/da7e052e0e43c5b3c286a572c6a84872b79cb4b5fa927e9056f6f2c1>
Requirement already satisfied: pandas in /usr/local/lib/python3.6/dist-packages (from pyspc) (0.25.3)
Requirement already satisfied: matplotlib in /usr/local/lib/python3.6/dist-packages (from pyspc) (3.1.2)

```
[2] # Producción de dulces en los Estados Unidos por fecha gestión y mes
#data = pd.read_csv("../Datos/candy_productions.csv")
data = pd.read_csv("candy_productions.csv")
```

```
[3] data.shape
```

```
(332, 6)
```

```
data.head(5)
```

	date	year	month	production1	production2	production3
0	1990-01-01	1990	1	99.9894	75.7999	142.7999
1	1990-02-01	1990	2	101.2116	73.7547	117.7547
2	1990-03-01	1990	3	94.8477	93.1582	108.1582
3	1990-04-01	1990	4	88.4239	102.2382	155.2382
4	1990-05-01	1990	5	88.6775	69.2708	155.2708

```
[5] data.index.dtype
```

```
dtype('int64')
```

```
[6] data1 = data.groupby('year').agg({'production1' : lambda x: x.sum(),
                                   'production2' : lambda x: x.sum(),
                                   'production3' : lambda x: x.sum()})

#data.columns = ['production1', 'production2']
data1.head()
```

	production1	production2	production3
year			
1990	1265.9570	1283.7596	1591.7596
1991	1316.8270	1499.4401	1309.4401
1992	1231.8878	1267.2503	1399.2503
1993	1255.2003	1364.3083	1360.3083
1994	1255.1259	1425.5262	1431.5262

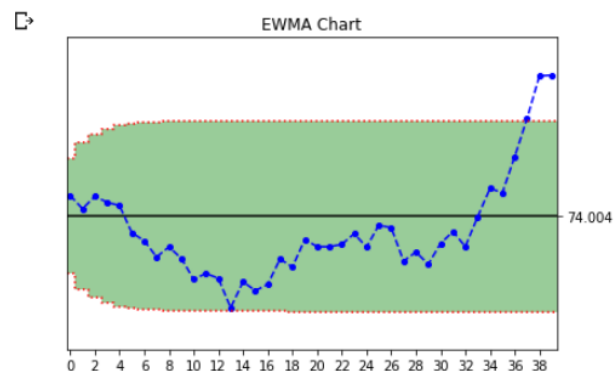

```
data.to_records().dtype
data.index = data.index.astype('i8')
cols = ['production1', 'production2', 'production3'] # Columnas que queremos cargar
data1[cols].head()
```

	production1	production2	production3
year			
1990	1265.9570	1283.7596	1591.7596
1991	1316.8270	1499.4401	1309.4401
1992	1231.8878	1267.2503	1399.2503
1993	1255.2003	1364.3083	1360.3083
1994	1255.1259	1425.5262	1431.5262

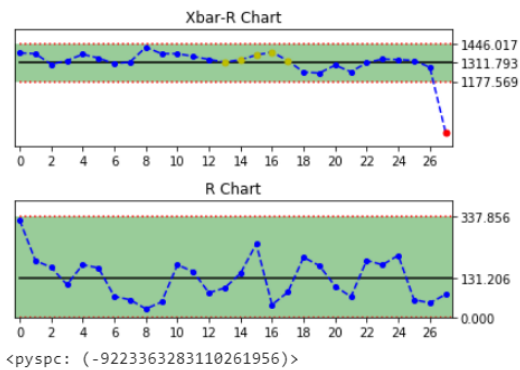
```
[8] #Datos desde 01-1990 hasta 08-2017
data2 = data1[cols].values
data2
#Datos desde 01-1990 hasta 08-2017
[8] data2 = data1[cols].values
data2
```

```
array([[1265.957 , 1283.7596, 1591.7596],
       [1316.827 , 1499.4401, 1309.4401],
       [1231.8878, 1267.2503, 1399.2503],
       [1255.2003, 1364.3083, 1360.3083],
       [1255.1259, 1425.5262, 1431.5262],
       [1270.1582, 1327.294 , 1435.294 ],
       [1322.318 , 1258.5922, 1328.5922],
       [1347.9634, 1289.318 , 1313.318 ],
       [1425.1335, 1404.9037, 1432.9037],
       [1396.5543, 1383.479 , 1345.479 ],
       [1462.279 , 1378.3539, 1286.3539],
       [1446.6292, 1293.7826, 1326.7826],
       [1385.3818, 1313.5118, 1303.5118],
       [1379.4629, 1283.2887, 1281.2887],
       [1394.5927, 1354.3612, 1247.3612],
       [1439.4358, 1205.456 , 1451.456 ],
       [1380.4629, 1368.0427, 1410.0427],
       [1300.7343, 1379.8707, 1295.8707],
       [1186.188 , 1173.3889, 1374.3889],
       [1133.7732, 1306.6877, 1272.6877],
```

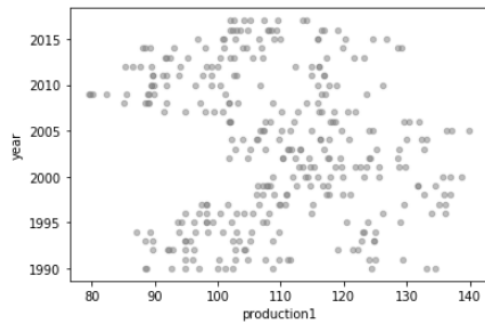
```
[9] a = spc(pistonrings) + ewma()
print(a)
```



```
[10] #data= data.values
a = spc(data2) + xbar_rbar() + rbar() + rules()
print(a)
```



```
[11] # Diagrama de dispersión produccion por año
var = 'production1'
data = pd.concat([data['year'], data[var]], axis=1)
data.plot.scatter(x=var, y='year', alpha = 0.5);
```



SEGUNDA PARTE

Aplicación de ciencia de datos (algoritmo de AI) a CADENA de SUMINISTROS.

Aplicación de sistema inteligente para implementar el modelo de gestión triple "A" en la cadena de suministro de centros de acopio de leche cruda

A. Descripción del problema:

Los procesos (**Recolección y acopio de leche, Pago a proveedores, Distribución y ventas**) que intervienen en la gestión de cadena de suministro en el centro de acopio de leche se detectan los problemas de tiempo y dinero.

Recolección y acopio de leche es el proceso donde los centros de acopio recolectan la leche en camiones, en la mayoría de los casos pertenecen a cada centro de acopio, los choferes tienen sueldos superiores a los 1.000 \$us. **no existe una ruta establecida** para realizar la recolección incurriendo en gastos innecesarios de tiempo, dinero y recursos.

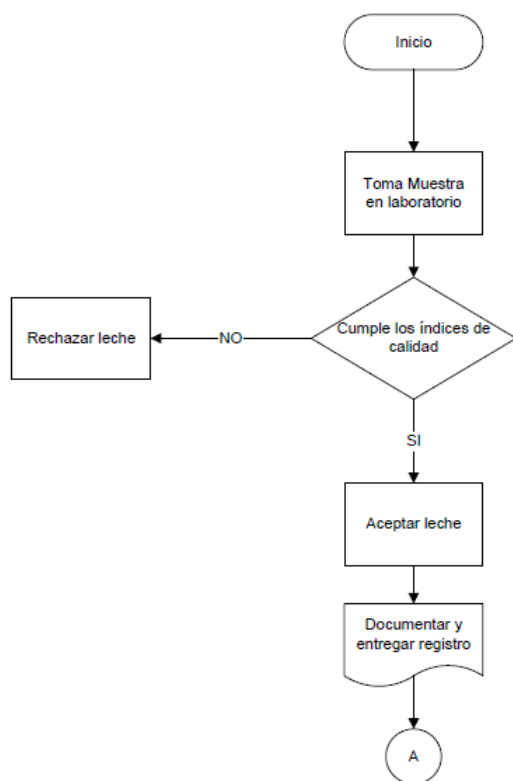


Fig.1 Recolección de leche

Fuente: Autores Tesis

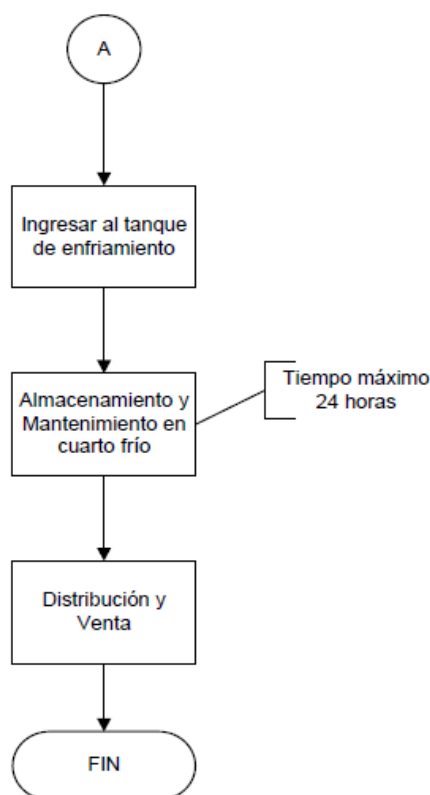


Fig.2 Almacenamiento y distribución

Fuente: Autores Tesis

B. Project Charter:

Project No.: 1		Date Charted: 10/12/2019		Rev. No.:		Rev. Date:					
APLICACIÓN DE SISTEMA INTELIGENTE PARA IMPLEMENTAR EL MODELO DE GESTION TRIPLE "A" EN LA CADENA DE SUMINISTRO DE CENTROS DE ACOPIO DE LECHE CRUDA											
Problema:				Objetivo del Proyecto							
Los procesos (Recolección y acopio de leche, Pago a proveedores, Distribución y ventas) que intervienen en la gestión de cadena de suministro en el centro de acopio de leche se detectan los problemas de tiempo y dinero. Recolección y acopio de leche , recolectan la leche en camiones, en la mayoría de los casos pertenecen a cada centro de acopio, <i>no existe una ruta establecido, esto</i> incurre en gastos innecesarios de tiempo, dinero y recursos.				En el caso del proceso de Recolección y acopio de leche el objetivo es la aplicación de sistemas inteligentes (Algoritmos genéticos) para la optimización de rutas de acopio de leche.							
Líder del proyecto				Enfoque del proyecto							
Lic. Juan López				ERP							
Grupo del proyecto				Personal de apoyo							
Name		Role		Name		Role					
Juna López		Contador		Carlos céspedes		Patrocinador					
Mario Camacho		Responsable recursos humanos		Mario Camacho		Gerente					
Noelia Ruiz		Responsable Ventas		Lucy Vera		Asistente contable					
Pablo Ortiz		Control de calidad		Fernando Arce		técnicos control calidad					
Ana Lujan		Logística		Lucas López y otros		Transportista					
Ana Méndez		Promoción Ventas									
Clientes y necesidades				Defectos							
Procesadoras de productos lácteos: Industria Lácteas Toni, Indulac, Nutrileche, Rey Leche, Otros.											
Métricas claves				Estimación de beneficios financieros							
Toma de muestra en laboratorio				Reducción de tiempo de traslado de leche							
Índices de calidad (análisis bromatológico)				Disminuir costo de transporte							
tiempo de transporte a centro de acopio				Reducir costos de mantenimiento y reparación de vehículos							
Costo de transporte de la leche											
Alcance del proyecto				Otra información del proyecto							
La aplicación se realizara a los procesos:											
Recolección y acopio de leche (caso de estudio)											
Pago a proveedores											
Distribución y ventas											
Fecha Inicio				Fecha Fin estimada							
23/12/2019				25/05/2020							
Hitos:		1		2		3		4		5	
Descripción:		Inicio del proyecto		Análisis		Implementación		Evaluación		Fin del proyecto	

Fecha:					
Signatures - The signatures of the people below document approval of the formal Project Charter					
	Firma			Fecha	
Líder del proyecto: Lic. Juan López				18/12/2019	
Patrocinador: Carlos Céspedes				18/12/2019	
Dueño del Proceso:				18/12/2019	
Asesor Financiero: Juan López				18/12/2019	

C. Stakeholder Analysis

APLICACIÓN DE SISTEMA INTELIGENTE PARA IMPLEMENTAR EL MODELO DE GESTION TRIPLE "A" EN LA CADENA DE SUMINISTRO DE CENTROS DE ACOPIO DE LECHE CRUDA									
Apoyo	Departmen to	Rol	Expectativa	Cómo son afectados	Poder(Nivel autoridad)	Interés (Nivel participación)	Importancia del Proyecto	Estrategia	Paso siguiente
Juna López	Finanzas	Contador	Reducir costos	aumenta la utilidad	8	4	8	Contacto semanal	Reunión de avances
Mario Camacho	RR.HH.	Responsable RR.HH.	Ajuste de personal	disminución de personal	7	5	7	Contacto mensual	Reunión de avances
Noelia Ruiz	Comercial	Responsable Ventas	ampliar demanda del mercado	mas ventas	7	7	6	Contacto semanal	Reunión de avances
Pablo Ortiz	Laboratorio	Control de calidad	mejorar control de calidad	menos desperdicio de leche	5	4	9	Contacto mensual	Reunión de avances
Ana Lujan	Logística	Logística	cumplir con las tareas asignadas	disminución de tiempo perdido	3	6	8	permanente contacto	Reunión de avances
Ana Méndez	Marketing	Promoción Ventas	Recuperar niveles de ventas	ganancias	5	3	6	Revisión de promociones	Reunión de avances
En contra	Departmen to	Rol	Expectativa	Cómo son afectados	Pder(Nivel autoridad)	Interés (Nivel participación)	Importancia del Proyecto	Estrategia	Paso siguiente
Lucas López	Logística	Transportista	quitaran rutas innecesarias	reducción de horas de trabajo	3	3	5		programar citas con Gerencia
Neutral	Departamento	Rol	Expectativa	Cómo son afectados	Poder (Nivel autoridad)	Interés (Nivel participación)	Importancia el proyecto	Estrategia	Paso siguiente
Carlos céspedes		Patrocinador	Reducción de costos		8	8	9		Reunión constante
Mario Camacho		Gerente	Mejorar ingresos	aumento de ingresos	10	5	10		Reunión de información
Lucy Vera	Finanzas	Asistente contable	Mejorar ingresos	aumento de ingresos	4	4	6		
Fernando Arce	Laboratorio	técnicos C.A.	Mejorar ingresos	aumento de ingresos	4	4	6		
Clientes			mejorar calidad	Aumento de acopio	2		4		Reunión

D. Gantt Chart:

Select a period to highlight at right. A legend describing the charting follows.

Period Highlight: 10

Plan Duration

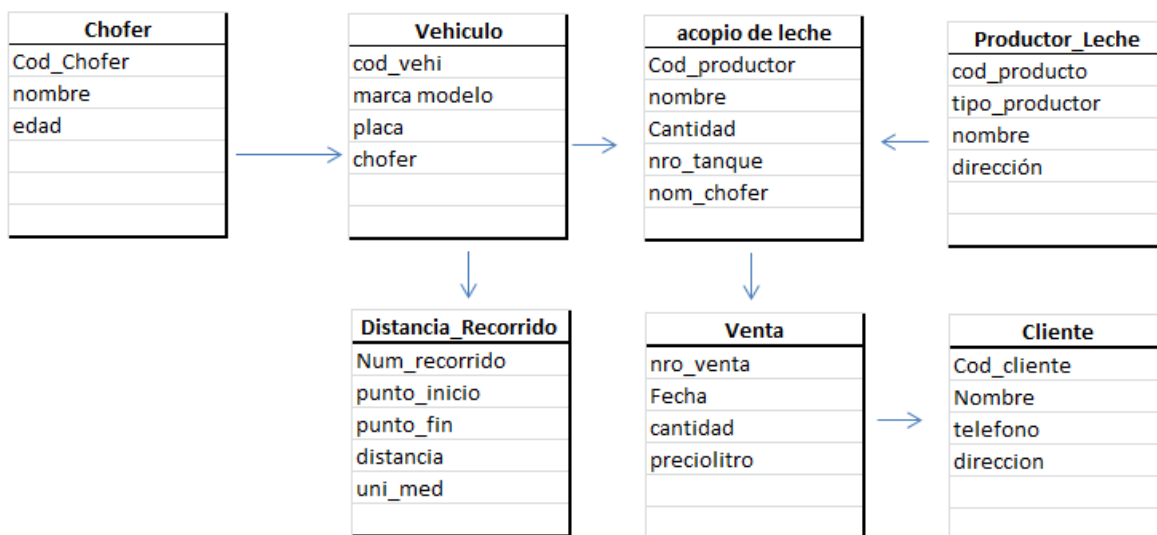
Actual Start

■ % Complete

Actual (beyond plan)

[illegible]

E. Arquitectura de los Datos y detalle de los datos:



Chofer: en el mayor de los casos es personal del centro de acopio, esta información está registrada en files físicos del departamento de recursos humanos.

Vehículo: es activo fijo del centro de acopio, lo tiene registrados en la parte de finanzas.

Acopio de leche: Son registros manuales de la recolección de leche, el chofer es el encargado de recolectar esta información.

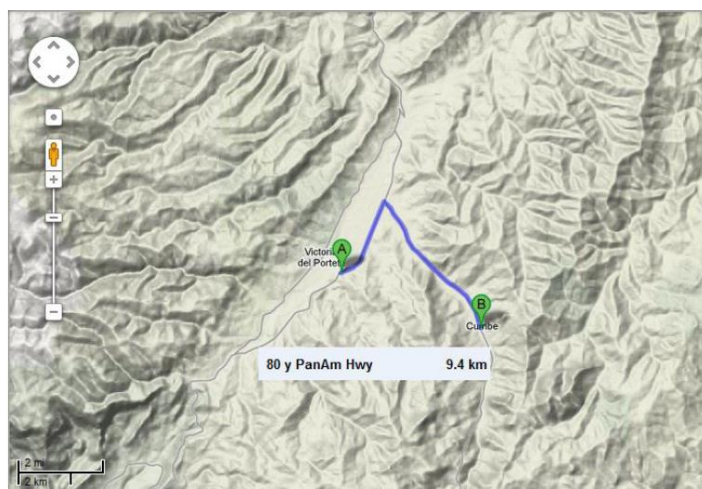
Productor de Leche: En laboratorio se tiene registrados a los productores de leche.

Ventas: Este datos se tiene en el departamento de contabilidad.

Cliente: Son las procesadoras de producto lácteo. Que se tiene en registros de contabilidad

Distancia de recorrido: Este dato se obtiene a través de google maps.

Medición de distancia del recorrido se obtiene a través de la página maps.google.com



Distancia entre dos puntos medida por carretera (google maps)

F. Detalle del algoritmo

Un **Algoritmo Genético** es una técnica metaheurística que es aplicada en problemas de búsqueda y **optimización**. Está basada en el proceso de Evolución y Selección Natural de las especies.

Para los cinco puntos del ejemplo, las distancias son en kilómetros según se indica el siguiente cuadro. El punto A corresponde al centro de acopio

Km.	A	B	C	D	E
A		9.4	15.2	10.4	14.3
B	9.4		20	11.4	7.2
C	15.2	20		7.8	10.7
D	10.4	11.4	7.8		6.5
E	14.3	7.2	10.7	6.5	

Distancia en kilómetros entre el centro de acopio y los puntos de recolección

Los pasos que cumple son:

1. Codificar en genes y cromosomas la información del problema

Para el caso de estudio se considera 5 puntos que tienen que ser recorridos, al ser cada tramo un origen y un destino, se necesitan cinco combinaciones tomadas de dos en dos, da como resultado diez bits para definir una trayectoria completa como se ve a continuación. (Recorrido del sistema de acopio por tramos)

AB	BC	CD	DE	EA	AC	BD	AD	BE	CE
Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1

Codificación del recorrido, sistema de acopio por tramos

2. Definir la función de adaptación

-Función evaluación es la combinación de la función $g(w)$ y $h(w)$

$$f(W) = g(W) - 0.1 * h(w)$$

Ec. 1

g(w) =Asegura que todos los N puntos del recorrido sean considerados.

h(w) =Utilizada para minimizar el costo (distancia) del recorrido o costo promedio.

Ec. 2
$$g(W) = 1 - \frac{|R - \sum_{i=0}^{L-1} b_i|}{R}$$

Ec. 3
$$h(W) = \frac{\sum_{i=0}^{L-1} b_i \cdot D_i}{\sum_{i=0}^{L-1} D_i}$$

Donde:

R: es el número de puntos de recorrido

L: cantidad de bit utilizados, para el ejemplo L=10

b_i: i-esímo bit del byte “W”

D_i: Coste/ distancia del tramo i

- **Función de aptitud A(W)** Es la función normalizada de la función de evaluación, sobre el total de la sumatoria de las funciones de evaluación de todos los **N** elementos de la población.

Ec. 4
$$A(W_k) = \frac{f(W_k)}{\sum_{i=0}^{N-1} f(W_i)}$$

3. Generar una población base

En este proceso, se define aleatoriamente una población que represente a la mayor cantidad de zonas. Para el caso de estudio se generará una población inicial de diez individuos.

Byte	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1
W1	0	0	1	1	0	1	1	1	0	1
W2	0	1	0	1	0	0	0	1	0	0
W3	0	0	0	1	0	0	1	1	1	1
W4	0	1	1	1	1	0	1	1	0	0
W5	1	0	1	1	1	0	0	0	1	0
W6	0	1	0	0	0	0	0	1	1	0
W7	0	0	0	0	0	0	1	0	1	1
W8	1	1	1	0	1	0	0	0	1	1
W9	0	0	0	1	0	0	0	0	0	1
W10	0	0	1	1	0	1	1	1	0	0

4. Evaluación y Selección

Para la evaluación sintetizamos la información, con la finalidad de facilitar el cálculo de la función de aptitud A(W)

Tramo	AB	BC	CD	DE	EA	AC	BD	AD	BE	CE
Byte W	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1
Distancia	9.4	20	7.8	6.5	14.3	15.2	11.4	10.4	7.2	10.7

Relación de Costo/Distancia por Bit y tramo

Con la población inicial y las ecuaciones 1,2,3 y 4 se puede calcular la función de aptitud de cada elemento de la población.

Byte	g(W)	h(W)	f(W)	A(W)	A(W) acumulada
W1 ♦	0.8	0.55	0.75	0.10	0.10
W2	0.6	0.33	0.57	0.08	0.18
W3 ♦	1	0.41	0.96	0.13	0.32
W4	0.8	0.62	0.74	0.10	0.42
W5 ♦	1	0.40	0.96	0.13	0.55
W6	0.6	0.33	0.57	0.08	0.63
W7	0.6	0.26	0.57	0.08	0.71
W8	0.8	0.61	0.74	0.10	0.81
W9	0.4	0.15	0.38	0.05	0.87
W10 ♦	1	0.45	0.95	0.13	1.00
		Total	7.19	1.00	

Evaluación de la población inicial

5. Reproducción

Para la reproducción se utilizará el **cruce por punto fijo** en el bit 6, por ejemplo seleccionamos los bytes W5 y W3 como padres se tendrían dos posibles descendencias.

Byte	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1
W5	1	0	1	1	1	0	0	0	1	0
W3	0	0	0	1	0	0	1	1	1	1
W5-W3	1	0	1	1	1	0	1	1	1	1
W3-W5	0	0	0	1	0	0	0	0	1	0

Cruce por punto fijo

Para el análisis se utilizarán como padres los mejores cuatro adaptados (W5, W3 W10 y W1)

Byte	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1
W5-W3	1	0	1	1	1	0	1	1	1	1
W3-W5	0	0	0	1	0	0	0	0	1	0
W5-W10	1	0	1	1	1	1	1	1	0	0
W5-W1	1	0	1	1	1	1	1	1	0	1
W3-W10	0	0	0	1	0	1	1	1	0	0
W3-W1	0	0	0	1	0	1	1	1	0	1
W10-W5	0	0	1	1	0	0	0	0	1	0
W1-W5	0	0	1	1	0	0	0	0	1	0
W10-W3	0	0	1	1	0	0	1	1	1	1
W1-W3	0	0	1	1	0	0	1	1	1	1

Segunda Generación de individuos

Ahora que se tiene la generación de padres e hijos se aplica la evaluación y selección de los diez mejores, con el mayor valor de la función de adaptación

Byte	g(W)	h(W)	f(W)	A(W)	A(W) acumulada
W5	1	0.40	0.96	0.12	3.60
W3	1	0.41	0.96	0.12	3.72
W10	1	0.45	0.95	0.11	3.83
W3-W1	1	0.48	0.95	0.11	3.95
W3-W10	0.8	0.39	0.76	0.09	4.04
W10-W3	0.8	0.48	0.75	0.09	4.13
W1-W3	0.8	0.48	0.75	0.09	4.22
W1	0.8	0.55	0.75	0.09	4.31
W8	0.8	0.61	0.74	0.09	4.40
W4	0.8	0.62	0.74	0.09	4.49
		Total	8.31	1.00	

Individuos que pasan a la siguiente generación

Se tiene como valor total de adaptación de 8.31 es superior al valor total de adaptación de la población inicial (7.19) Lo que indica que esta generación es más apta que la inicial.

6. Mutación

Puede ser combinada con la reproducción, ejecutada aleatoriamente o aplicada luego de que en varias generaciones no se produzca un crecimiento en la función de aptitud.

7. Condición de la terminación

Se define por la cantidad de ciclos o cuando la función de aptitud alcance un valor meta, en nuestro ejemplo se realizó en dos ciclos con el fin de modelar un caso práctico de optimización de rutas mediante algoritmos genéticos.