

# gestioninventarios.pdf



**Cristina\_Acosta**



**Ampliación de Investigación Operativa**



**3º Grado en Estadística**



**Facultad de Matemáticas  
Universidad de Sevilla**

**Correcto.**

**¿ES CON V O CON B?**  
Con C, de Correcto.

**¡Pruébalo aquí!**



#escribeunfuturomejor



MÁNCHATE LOS DEDOS CON CHEETOS  
Y ESCAQUÉATE  
DE TOMAR APUNTES HOY



## Gestion de Inventarios

Cristina Acosta Muñiz

### Índice

<b>1. Introducción</b>	<b>2</b>
<b>2. Parametros básicos</b>	<b>2</b>
2.1. Costes asociados	2
2.2. Demanda	2
2.3. Plazo de entrega	2
2.4. Otros factores	3
<b>3. Objetivo</b>	<b>3</b>
<b>4. Tipos de Stocks</b>	<b>3</b>
<b>5. Gestion de Inventarios en R</b>	<b>3</b>
5.1. Ejercicio flanes	3
5.2. Modelo EPQ	4
5.3. Modelo Newsboy	5
5.4. Punto de reposicionamiento (ROP)	6
5.5. Nivel de seguridad (SS)	7
5.6. Método WW	7

## 1. Introducción

El inventario de un almacén incluye bienes y materiales usados en la logística de servicios de una empresa.

- Se necesita para reducir la falta de suministro de productos a clientes
- Se establece un equilibrio entre la calidad en el suministro de productos e inversión necesaria
- Gastos e inversiones derivadas de un inventario:
  - Gastos del capital invertido (intereses, impuestos, seguros, ...)
  - Espacio, mano de obra y medios de transporte
  - Deterioro por obsolescencia, por robo o por ser material perecedero

## 2. Parametros básicos

### 2.1. Costes asociados

- Coste de pedido ( $C_p$ ).
  - Aspectos administrativos
  - En caso de fábricas, el coste de montaje
  - Expresado en unidades monetarias/pedido
- Coste de almacenamiento ( $C_s$ ).
  - Costes de capital, seguros, impuestos, robos y deterioros
  - Obsolescencia
  - Amortización de almacenes y personal
  - Expresado como una tasa porcentual de almacenamiento anual  $k$
- Coste de ruptura o penuria ( $C_r$ ).
  - Costes indirectos de falta de suministro de demanda a clientes
  - Expresado en unidades monetarias/tiempo/producto

### 2.2. Demanda

La demanda se pueden aplicar distintos modelos:

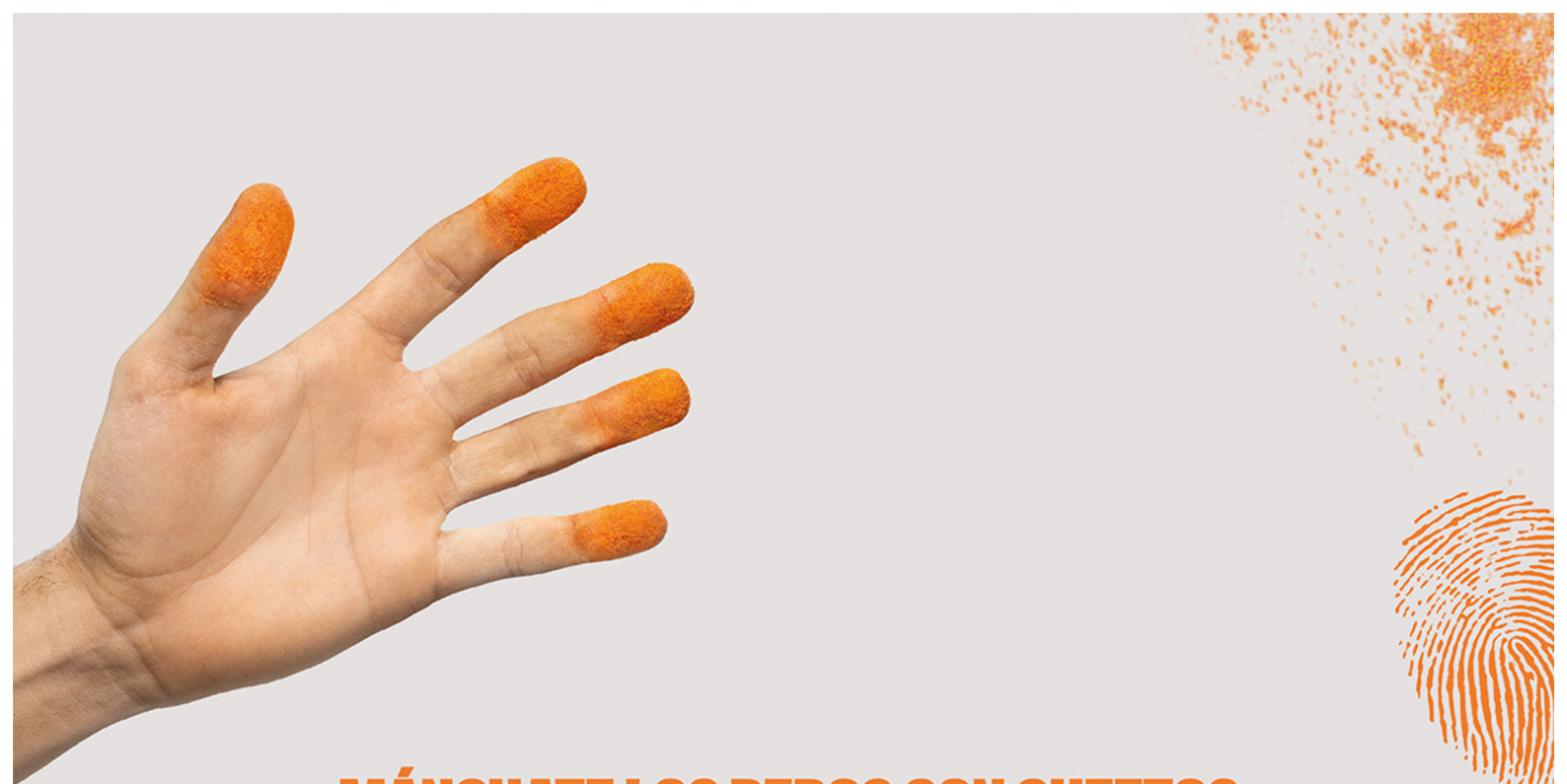
- valor discreto o continuo
- valor determinista o aleatorio
- valor independiente, dependiente o mixto

### 2.3. Plazo de entrega

El plazo de entrega es el tiempo entre la emisión del pedido y la recepción:

- valor determinista o aleatorio





**MÁNCHATE LOS DEDOS CON CHEETOS**  
**Y ESCAQUÉATE**  
**DE TOMAR APUNTES HOY**



## 2.4. Otros factores

- comportamiento proveedores (tamaño de pedido limitado, periodicidad,...)
- requerimientos de los pedidos (calidad de suministro,...)
- ciclo de vida del producto (productos perecederos)

## 3. Objetivo

Establecer un equilibrio entre la calidad del servicio y el coste económico de la gestión de stocks.

En este tipo de problemas siempre hay que decidir: **¿Cuándo?** y **¿Cuanto?** que son las variables de decisión que van a determinar una función objetivo que vamos a minimizar si son costos.

## 4. Tipos de Stocks

- **Stock EN CURSO** ( $S_C$ ): Aquél que ha sido pedido pero no ha llegado aún
- **Stock ASIGNADO** ( $S_A$ ): Aquél que está en el almacén y ha sido comprado
- **Stock FÍSICO** ( $S_F$ ): Aquél que está en el almacén
- **Stock LOGÍSTICO** ( $S_L$ ): Suma del stock físico y del stock en curso
- **Stock DISPONIBLE** ( $S_D$ ): Aquél que está en el almacén y no ha sido asigna

$$S_L = S_C + S_D = S_C + S_F - S_A$$

## 5. Gestion de Inventarios en R

### 5.1. Ejercicio flanes

Para resolver los ejercicios de gestión de inventarios vamos a usar la librería **SCperf**.

```
library(SCperf)
```

Modelo sin rotura de stock

Datos:

```
demanda <- 500000 #Producción de flanes al año
Te <- 1 # Tiempo total (en años)
Cp <- 300 # Coste de pedido en euros por pedido
Cs <- 9/100*0.3 # Coste de almacenamiento anual: el 30% de 9 centimos (coste de adquisición)
```

Modelo:

```
modelo1 <- EOQ(d = demanda, k = Cp, h = Cs) # sin ruptura de stock
modelo1
```

```
##          Q          T          TVC
## 105409.26      0.21    2846.05
```

Solución:

- $Q$  = Cantidad a pedir (105409.26)
- $T$  = Cada cuanto tiempo pedir (0.21 años = 2.5 meses)
- $TVC$  = coste anual de la gestión (2846.05 €)

### Modelo con rotura de stock

```
Cr <- 0.2 # Coste de ruptura (0.20€ por envase/año)
modelo2 <- EOQ(d = demanda, k = Cp, h = Cs, b = Cr) # con ruptura de stock
modelo2
```

```
##          Q          T          S          TVC
## 112299.20      0.22  13357.17   2671.43
```

Solución:

- Q = Cantidad a pedir (112299.20)
- T = Cada cuanto tiempo pedir (0.22 años = 2.6 meses)
- S = maximo pedidos en unidades(13357.17)
- TVC = coste anual de la gestión (2671.43 €)

## 5.2. Modelo EPQ

En este modelo no nos llega de golpe todo lo que necesitamos, es decir, en este modelo la capacidad de fabricación de la fabrica, al proveedor que le pedimos los envases no es capaz de enviarlos todos a la vez y lo hara poco a poco. Entonces, la tasa de producción debe ser mayor que la demanda.

Tenemos que:  $EPQ(d, p, k, h, b = 0)$

- d: demanda
- p: tasa de produccion
- k: coste de pedido
- h: coste de almacenamiento
- b: penalizacion

La salida que nos devuelve seria:

- Q: cuanto debemos pedir.
- t: el tiempo que van a tardar en producir
- T: cada cuanto tiempo debemos pedir
- I: Maximo nivel de inventario
- TC: Coste total

### 5.2.1. Ejercicio

1. ¿Qué pasaria en un modelo de producción no instantanea con un modelo EPQ si en vez de producir instantaneamente 12000 que le pedimos, tenemos una fabrica que es capaz de producir al año 1000000 de envases?

Datos:

```
demanda <- 500000 #Producción de flanes al año
tasa_prod <- 1000000 #Tasa de produccion en años
Cp <- 300 # Coste de pedido en euros por pedido
Cs <- 9/100*0.3 # Coste de almacenamiento anual: el 30% de 9 centimos (coste de adquisición)
```

Modelo:

```
modelo3 = EPQ(d = demanda, p=tasa_prod, k = Cp, h = Cs) #sin rotura de stock
print(round(modelo3, 2))
```



MÁNCHATE LOS DEDOS CON CHEETOS  
Y ESCAQUÉATE  
DE TOMAR APUNTES HOY



##	q	t	T	I	TC
## 149071.20		0.15	0.30	74535.60	2012.46

Solución:

- q: La cantidad que debemos de pedir es 149071.20 unidades
- t: Va a tardar en producir 0.15 años (1.8 meses)
- T: Debemos de pedir cada 0.30 años (3.6 meses)
- I: Podemos tener como maximo almacenado 74535.60 unidades
- TC: El coste total del proceso es 2012.46

2. ¿Y que pasaría si hubiera costes de rotura?

Datos:

```
demanda = 500000
tasa_prod = 1000000
Cp = 300
Ca = 9/100*0.3
ruptura = 20/100
```

Modelo:

```
modelo4 = EPQ(d = demanda, p=tasa_prod, k = Cp, h = Ca, b = ruptura)
modelo4
```

##	q	t	T	I	TC
## 2.13073e-01	2.13073e-07	4.26146e-07	1.06536e-01	7.03985e+08	

### 5.3. Modelo Newsboy

Se trata de un problema estocastico muy simple donde tenemos la media de la demanda y la desviación típica de la demanda donde vamos a obtener cuantos periodicos nos van a comprar. Esto nos sirve para cualquier producto.

El modelo del vendedor de periodico tiene los siguientes argumentos:

Newsboy(m, sd, p, c, s = 0) donde

- m: media de la demanda
- sd: desviación típica
- p: precio
- c: coste
- s: valor de reventa

Los valores que obtenemos son:

- Q: La cantidad que nos aconseja que compremos
- SS: Stock de seguridad
- ExpC: Costo esperado
- ExpP: Costo del beneficio
- CV: Coeficiente de variación de la demanda



- FR: Tasa de la demanda que va a ser servido si pedimos esa cantidad
- z: Factor de seguridad

Ejemplos:

```
Newsboy(100,30,4,1)
```

```
##      Q      SS  ExpC  ExpP      CV      CR      FR      z
## 120.23 20.23 38.13 261.87  0.30  0.75  0.96  0.67
```

```
Newsboy(100,20,4,1)
```

```
##      Q      SS  ExpC  ExpP      CV      CR      FR      z
## 113.49 13.49 25.42 274.58  0.20  0.75  0.97  0.67
```

### 5.3.1. Ejercicio

Con una demanda estimada de 150, una desviación típica de 30, una venta de 10 y la adquisición es de 5.

1. ¿Cuanto deberíamos de comprar al día para obtener la mejor política posible?

Datos:

```
dem_est <- 150
desv_tip <- 30
venta <- 10
adquisicion <- 5
```

Modelo:

```
Newsboy(m = dem_est, sd = desv_tip, p = venta, c = adquisicion)
```

```
##      Q      SS  ExpC  ExpP      CV      CR      FR      z
## 150.00  0.00 119.68 630.32  0.20  0.50  0.92  0.00
```

Solución:

Debemos de pedir 150 unidades ( $Q = 150$ ).

2. ¿Qué pasaría si pudieramos devolver el producto vendiendolo a 2 unidades manteniendo los datos unidades?

```
Newsboy(m = dem_est, sd = desv_tip, p = venta, c = adquisicion, s=2)
```

```
##      Q      SS  ExpC  ExpP      CV      CR      FR      z
## 159.56  9.56  91.01 658.99  0.20  0.62  0.95  0.32
```

Podemos ver que en vez de compra 150, nos podemos arriesgar a comprar 160 unidades para tener una política optima.

### 5.4. Punto de reposicionamiento (ROP)

Otra forma de enfocar el problema para valores estocasticos. Tenemos la función  $ROP(SL, md, sd, L = 1)$  donde

- SL: nivel de servicio que queremos tener
- md: demanda media
- sd: desviación típica de la demanda
- L: holgura de lo que queremos.



El valor que obtenemos es en que nivel de inventario debemos de volver a pedir, es decir, cuantos productos debe de pedir para no quedar sin stock en almacen con una probabilidad.

Ejemplos:

```
ROP(0.9,2500,500,6)
```

```
## [1] 16570
```

```
ROP(0.9,2500,500,1)
```

```
## [1] 3141
```

```
ROP(0.9,2500,500,10)
```

```
## [1] 27026
```

```
ROP(0.99,2500,500,10)
```

```
## [1] 28678
```

```
ROP(0.8,2500,500)
```

```
## [1] 2921
```

```
ROP(0.9,2500,500)
```

```
## [1] 3141
```

```
ROP(0.99,2500,500)
```

```
## [1] 3663
```

## 5.5. Nivel de seguridad (SS)

El nivel de seguridad lo podemos calcular con la función  $SS(SL, sd, L = 1)$  con los argumentos

- SL: nivel de servicio
- sd: desviación típica de la demanda
- L: tiempo de elaboración

El valor que nos devuelve es el nivel de stock de seguridad.

Ejemplo:

```
SS(0.95,0.7,2)
```

```
## [1] 1.6
```

```
SS(0.95,0.1,2)
```

```
## [1] 0.23
```

## 5.6. Método WW

Este método lo usaremos en caso de tener varias etapas.

Ejemplo:

```
x <- c(3,2,3,2) #Lo que nos piden que produzcamos, 10
a <- 2 #Costes de empezar a producir
h <- 0.2 #Costes de almacenamientos
WW(x,a,h,method="backward")
```

```
## Call:
## WW.default(x, a, h, method = "backward")
##
## TVC:
## [1] 4.8
##
## Solution:
##      [,1] [,2] [,3] [,4]
## [1,] 5.4 4.8 5.6 4.8
## [2,] NA 4.4 4.6 3.4
## [3,] NA NA 4.0 2.4
## [4,] NA NA NA 2.0
##
## Jt:
## [1] "2 or 4" "4" "4"
```

Nos dice que el coste total de este proceso es 4.8 ( $TVC = 4.8$ ), y ahora lo que nos dice abajo es lo que tenemos que producir en cada etapa para obtener el coste optimo.

En Jt nos dice que las soluciones posibles serian comenzar produciendo 2 o 4. Entonces, si producimos 4 lo que estamos diciendo es entregamos 3 y uno de ellos estara en el almacen en la primera etapa. En la segunda etapa, nos esta diciendo que lo razonable seria es producir 4.

El coste minimo es 4.8, este coste se encuentra en dos sitios: en la dila 2 y en la fila 4. Luego la solución 1 es producir todo lo que necesitamos hasta el periodo dos. Si es todo lo que necesitamos hasta el periodo dos, lo que nos esta diciendo es que produzcamos al principio 5 y luego en el periodo 3 debemos de volver a decidir cuanto vamos a fabricar. En este caso, lo mas razonable seria producir en el periodo 3 los 5 que quedan mientras que la solución 2, que se encuentra en fila de arriba y en la cuarta columna, producir los 10 simultaneamente.

Si lo vemos en la matriz del resultado:

Nos fijamos en la primera fila, nos tenemos que ir a la columna 2 o 4. Si elegimos la 4 nos dice que produzcamos todo de golpe. Entonces si produzco todo hasta el final, entonces producimos 10. (4.8 significa hacerlo todo hasta el final)

Sin embargo, en la segunda fila vemos que tenemos que producir lo de la primera y segunda etapa.

Si cambiamos los datos, por ejemplo:

```
x <- c(5,1,1,3) #10
WW(x,a,h,method="backward")
```

```
## Call:
## WW.default(x, a, h, method = "backward")
##
## TVC:
## [1] 4.4
##
## Solution:
##      [,1] [,2] [,3] [,4]
## [1,] 5.4 4.8 4.6 4.4
## [2,] NA 4.6 4.2 3.4
## [3,] NA NA 4.0 2.6
## [4,] NA NA NA 2.0
##
## Jt:
## [1] "4" "4" "4" "4"
```



# MÁNCHATE LOS DEDOS CON CHEETOS Y ESCAQUÉATE DE TOMAR APUNTES HOY



No hay duda de que siempre es la columna 4. La decision es que en la primera etapa hacerlo todo de golpe, es decir, si cogemos y de golpe hacemos los 10 esa es la politica que nos va a costar 4.4

Si volvemos a cambiar los datos:

```
x <- c(1,1,1,7) #10
WW(x,a,h,method="backward")

## Call:
## WW.default(x, a, h, method = "backward")
##
## TVC:
## [1] 4.6
##
## Solution:
##      [,1] [,2] [,3] [,4]
## [1,]  6.2  5.6  4.6  6.8
## [2,]   NA  5.4  4.2  5.0
## [3,]   NA   NA  4.0  3.4
## [4,]   NA   NA   NA  2.0
##
## Jt:
## [1] "3" "3" "4" "4"
```

En este caso, vemos que el optimo esta en el 3. Entonces es preferible cubrir las necesidades de los 3 primeros, es decir, solo fabricar 3 y luego fabricar en la ultima etapa los que nos quedan que serian 7.

```
x <- c(10,0,0,0)
a <- 2
h <- 0.3
WW(x,a,h,method="backward")

## Call:
## WW.default(x, a, h, method = "backward")
##
## TVC:
## [1] 2
##
## Solution:
##      [,1] [,2] [,3] [,4]
## [1,]    4    4    4    2
## [2,]   NA    4    4    2
## [3,]   NA   NA    4    2
## [4,]   NA   NA   NA    2
##
## Jt:
## [1] "4" "4" "4" "4"
```

Lo que nos dice es que lo logico es que nos vayamos al coste menor que esta en la posicion 4, es decir, en cada una de las filas el coste minimo se encuentra en la poscisión 4.

El 2 en la poscicion 4 significa que debemos de fabricar todo de golpe. Si me piden 10, fabrico los 10. Lo que me va a costar fabricar los 10 sera 2 porque no tengo ningun gasto de almacenamiento.

El 2 sale de que vamos a tener un coste de fabricación único y por tanto no hay ningun coste de almacenamiento porque me han pedido 10 al principio y los he entregado.

```
x <- c(5,5,0,0)
WW(x,a,h,method="backward")
```

```
## Call:
## WW.default(x, a, h, method = "backward")
##
## TVC:
## [1] 3.5
##
## Solution:
##      [,1] [,2] [,3] [,4]
## [1,]    4  5.5  5.5  3.5
## [2,]   NA  4.0  4.0  2.0
## [3,]   NA   NA  4.0  2.0
## [4,]   NA   NA   NA  2.0
##
## Jt:
## [1] "4" "4" "4" "4"
```

Ahora el mínimo coste es 3.5 y se encuentra en la cuarta posición. Este 3.5 significa que si lo fabricamos todo de golpe me va a costar 2 unidades. El  $1.5(=3.5-2)$  que queda significa que durante un periodo de tiempo tenemos 5 unidades almacenadas del segundo periodo porque fabrico primero 5 unidades en el primer periodo y entrego esas 5 unidades al cliente. Que almacenar esas 5 unidades me va a costar  $0.3*5=1.5+2=3.5$ , esta es la solución óptima. Es la solución óptima porque si quisiéramos otras políticas lo que nos iría dando un coste mayor de almacenamiento o bien de producción. Solo debemos interpretar la solución óptima, que significa.

Interpretación:

La primera columna de la primera significa fabricar solo lo que nos pidan hasta la primera etapa, es decir, fabricar 5 hasta la primera etapa. La interpretación de la segunda columna sería hacer en la primera petición lo que nos piden en la primera y en la segunda. En este caso, 5 y 5. La tercera columna sería hacer lo que nos pide en la primera, en la segunda y en la tercera. La cuarta columna sería hacerlo todo de golpe. Esto sería la estructura de los valores.

La solución óptima se encuentra en el coste mínimo que es 3.5, que corresponde a la cuarta columna que es fabricar todo de golpe al principio. Si fabricamos todo de golpe la producción fija nos va a costar 2, es decir, solo vamos a gastar 2 unidades. Hasta 3.5 serían costes de almacenamiento, 0.3. Sería 0.3 porque estamos fabricando 5 unidades que las tendríamos que guardar durante un periodo y cada uno de esos periodos nos cuesta  $0.3*5=1.5+2=3.5$ .

Nota: Solo nos debemos de fijar en la primera fila si la solución óptima se encuentra en la última columna.

```
x <- c(1,1,1,1)
WW(x,a,h,method="backward")
```

```
## Call:
## WW.default(x, a, h, method = "backward")
##
## TVC:
## [1] 3.8
##
## Solution:
##      [,1] [,2] [,3] [,4]
## [1,]  4.9  4.6  4.9  3.8
## [2,]   NA  4.3  4.3  2.9
## [3,]   NA   NA  4.0  2.3
## [4,]   NA   NA   NA  2.0
```

```
##
## Jt:
## [1] "4" "4" "4" "4"
```

Nos esta diciendo que con estos costes sigue siendo mas rentable producir todo de golpe.

```
x <- c(1,1,1,1)
a <- 0.2
h <- 3
WW(x,a,h,method="backward")
```

```
## Call:
## WW.default(x, a, h, method = "backward")
##
## TVC:
## [1] 0.8
##
## Solution:
##      [,1] [,2] [,3] [,4]
## [1,] 0.8 3.6 9.4 18.2
## [2,] NA 0.6 3.4 9.2
## [3,] NA NA 0.4 3.2
## [4,] NA NA NA 0.2
##
## Jt:
## [1] "1" "2" "3" "4"
```

En este caso, si debemos de fijarnos en todas las columnas. En la primera columna nos esta diciendo que el minimo se encuentra en la primera fila de la primera columna, por tanto solo produzco 1 en la primera etapa.

Nos vamos a la fila 2, el minimo esta en 0.6. Por tanto, solo debemos producir lo que nos pidan para esa etapa.

Nos vamos a la fila 3, el minimo se encuentra en la columna 3. Por tanto, solo debemos producir lo que nos pidan para esa etapa.

Y por último, el ultimo valor significa que produzcamos lo que pidan en la ultima etapa.

Luego, nos esta diciendo que en cada etapa produzcamos una unidad donde el optimo es 0.8 y ademas no gasto nada de almacenaje porque fabrico y entrego al cliente.

```
x <- c(1,1,5,1)
a <- 0.2
h <- 3
WW(x,a,h,method="backward")
```

```
## Call:
## WW.default(x, a, h, method = "backward")
##
## TVC:
## [1] 0.8
##
## Solution:
##      [,1] [,2] [,3] [,4]
## [1,] 0.8 3.6 33.4 42.2
## [2,] NA 0.6 15.4 21.2
## [3,] NA NA 0.4 3.2
## [4,] NA NA NA 0.2
```



```
##
## Jt:
## [1] "1" "2" "3" "4"
```

Obtenemos lo mismo porque como el coste de almacenaje es alto, otra politica no seria posible.

Conclusion, nos esta diciendo que fabriquemos y entreguemos por tener los costes de almacenamientos tan altos.

```
x <- c(1,1,5,1)
a <- 2
h <- 0.3
WW(x,a,h,method="backward")
```

```
## Call:
## WW.default(x, a, h, method = "backward")
##
## TVC:
## [1] 4.6
##
## Solution:
##      [,1] [,2] [,3] [,4]
## [1,]  6.1  4.6  7.3  6.2
## [2,]   NA  4.3  5.5  4.1
## [3,]   NA   NA  4.0  2.3
## [4,]   NA   NA   NA  2.0
##
## Jt:
## [1] "2" "4" "4" "4"
```

Nos dice que nos fijemos en la columna 2. Primero se producen las dos primeras peticiones y luego nos tenemos que ir a la file 3 donde se encuentra el NA e irnos al minimo que es 2.3. Por tanto, me es mas rentable producir los 6 que me quedan en la tercera etapa.

La solución seria fabricar en la primera etapa 2 y en la tercera etapa 6.

```
a <- 2
h <- 0.3
x <- c(6,1,5,1,2,7)
WW(x,a,h,method = "backward")
```

```
## Call:
## WW.default(x, a, h, method = "backward")
##
## TVC:
## [1] 7.8
##
## Solution:
##      [,1] [,2] [,3] [,4] [,5] [,6]
## [1,]  9.5  7.8  9.9 10.2 10.6 19.1
## [2,]   NA  7.5  8.1  8.1  7.9 14.3
## [3,]   NA   NA  6.6  6.3  5.5  9.8
## [4,]   NA   NA   NA  6.0  4.6  6.8
## [5,]   NA   NA   NA   NA  4.0  4.1
## [6,]   NA   NA   NA   NA   NA  2.0
##
## Jt:
## [1] "2" "2" "5" "5" "5" "6"
```

# Hazte cliente de BBVA y ... ahórrate 6 meses de suscripción

WUOLAH  
+ BBVA

NETFLIX

Spotify

HBOmax

Disney+

PlayStation.Plus

DAZN

Ahora, si te abres una Cuenta Online en BBVA, te reembolsamos una de estas suscripciones durante 6 meses (hasta 9,99€/mes) al pagarla con tu tarjeta Aqua Débito

Promoción solo  
para nuevos  
clientes de  
BBVA. Válida  
hasta el  
30/06/2023.  
Estas empresas  
no colaboran  
en la  
promoción.

1/6

Este número es  
indicativo del  
riesgo del  
producto, siendo  
1/6 indicativo de  
menor riesgo y  
6/6 de mayor  
riesgo.

BBVA está  
adherido al  
Fondo de  
Garantía de  
Depósitos de  
Entidades de  
Crédito de  
España.  
La cantidad  
máxima  
garantizada es  
de 100.000 euros  
por la totalidad  
de los depósitos  
constituidos  
en BBVA por  
persona.

El minimo esta en 7.8.

Etapas 1 produzco 7 (min col 2: 7.8)

Etapas 2 produzco 8 (min col 5: 5.5)

Etapas 3 produzco 7 (min col 6: 2.0)



Abre tu cuenta



WUOLAH  
+ BBVA

Reservados todos los derechos. No se permite la explotación económica ni la transformación de esta obra. Queda permitida la impresión en su totalidad.