Algoritmo de Bernstein

1. Dada $\mathcal{U}$ (relación universal)
   $\mathcal{U}$ = [{a_i} , {df_k}]
   En el ejemplo $\mathcal{U}$ = [{l, n, e, s} , {l→n,e,s   e→s}]
2. Lado Derecho Simple (Reglas de Armstrong)
   a → b,c ⇒ a → b , a → c (r. descomposición)
   Tomar toda DF que tenga más de un atributo dependiente y aplicar la descomposición
   En el ejemplo partiendo de {l→n,e,s   e→s} nos queda
   l → n   l→e   l→s por descomp. y e→s
3. Encontrar Mínima Cobertura No Redundante MCNR.
   Aplicar Algoritmo de Cerradura. (x-Closure). Conjunto de atributos que son determinados luego de dar un atributo inicial.
   Ver funcionamiento de algoritmo
4. Lado Izquierdo mínimo
   a,x → y a → x ⇒ a → y   a → x
   En el ejemplo no se hace nada.
5. Encontrar Claves de $\mathcal{U}$ (Candidatas)
6. Reunir DFs con mismo lado izquierdo
   a → b a → c ⇒ a → b,c (r.aditividad)
7. Crear Tablas con cada DF (aquí se definen las PK)
   a → b,c   TABLA(<u>a</u>, b, c) estas tablas están en 3FN
8. Si alguna clave candidata obtenida en 5. NO aparece en las Tablas obtenidas en 7. se debe agregar una tabla para dicha clave.

   Algoritmo de Clausura (Cerradura) x-Closure

```
Parámetros de entrada ({DF_k} , CAtrib)
i ← 0; t ← TRUE; X[0] ←CAtrib;
Mientras t = TRUE Hacer:
|       T ← FALSE; j ← 0;
|       Para toda DF ∈ {DF_k} Hacer:
|       |       Si toda ( LI(DF) ∈ X[i]  Y
|       |                 LD(DF) ∉ X[i] )
|       |       Entonces
|       |                 j ← i + 1;
|       |                 X[j] ← X[i] U LD(DF);
|       |                 t ← TRUE; i ← i +1;
|       Next
Fin
Retornar (X[i]);
FIN
```

Funcionamiento del x-closure

Tomamos un determinante. Por ejemplo legajo (l)

X[0] = l	paso inicial 2da línea
X[1] = l, n	analiza 1°DF *l → n*
X[2] = l, n, e	analiza 2°DF *l → e*
X[3] = l, n, e, s	analiza 3°DF *l → s*
queda igual	analiza 4°DF *e → s*

luego sigue tomando el siguiente atributo, y así hasta el último
el resultado (cerradura) final es X[3] = l, n, e, s

Ejemplo 1

En la siguiente tabla la edad determina el sueldo.

EMPLEADOS

<u>Legajo</u>	***nombre***	***edad***	***sueldo***
8	José	50	5000
9	Ana	20	3000
10	Carlos	50	5000

Cumple 1FN y 2FN. No así la 3FN.

EMPLEADO

<u>Legajo</u>	***nombre***	***edad***
8	José	50
9	Ana	20
10	Carlos	50

3FN

SUELDO-E

<u>edad</u>	***Sueldo***
20	3000
50	5000

Cómo lograr que un algoritmo logre esto:

Dependencias Funcionales

legajo → nombre, edad, sueldo
edad → sueldo

l → n, e, s
e → s

El AB toma como entrada el conjunto de atributos $\{a_j\}$ y las DF $\{a_j \rightarrow b_j\}$

Esto se denomina Relación Universal: $\mathcal{U} = [\{a_j\}, \{a_j \rightarrow b_j\}]$