



## Índice

# Introducción

La motivación de este proyecto surge de una experiencia personal que despertó mi interés por comprender cómo el organismo regula los niveles de glucosa en la sangre. Durante un periodo en el que consumía una cantidad elevada de azúcar, me pregunté cómo funciona este proceso de regulación y, en particular, cómo las personas con diabetes dependen de la administración de insulina para mantener su glucemia dentro de rangos saludables.

A partir de esta inquietud, decidí desarrollar un modelo capaz de generar un plan de administración de insulina para personas con diabetes tipo 1, considerando la fluctuación temporal de la glucosa. Sin embargo, al iniciar la propuesta aún no tenía experiencia en Q-Learning, lo que condiciona tanto el enfoque como las dificultades que surgieron posteriormente.

Mi desconocimiento inicial sobre el funcionamiento de un agente de Q-Learning me llevó a realizar una investigación que resultó, en gran medida, innecesaria: construí una base de datos detallada con parámetros cuidadosamente diseñados para representar pacientes dentro de dos desviaciones estándar de variación. Al no tener acceso a datos médicos reales —por su carácter privado— consulté diversos artículos (citados en las referencias) para sustentar la creación de estos datos sintéticos. Finalmente, comprendí que Q-Learning solo requiere un estado inicial, una función de recompensa y un estado objetivo, por lo que la base de datos resultó irrelevante salvo por el valor inicial de insulina.

Posteriormente implementé el agente y lo entrené utilizando estos datos. En total, entrené 100 modelos de Q-Learning con 500 pacientes cada uno, seleccionando el mejor desempeño para aplicarlo al control de insulina de los 50.000 pacientes que conforman la base de datos. No obstante, dado que la motivación original del proyecto era diseñar un plan de administración de insulina para una persona específica, se incorporó una función que permite generar dicho plan utilizando el mejor modelo entrenado.

# Modelo escogido y cómo funciona

## ¿CÓMO FUNCIONA Q-LEARNING?

Q-Learning es un algoritmo de aprendizaje por refuerzo que aprende una política óptima mediante:

Tabla Q (Q-table): Matriz estados × acciones que almacena la calidad esperada (Q-value) de cada par estado-acción

Ecuación de Bellman: Actualiza Q-values:

$$Q(s,a) \leftarrow Q(s,a) + \alpha[R + \gamma \cdot \max(Q(s',a')) - Q(s,a)]$$

$\alpha$  (alpha): Tasa de aprendizaje

$\gamma$  (gamma): Factor de descuento

R: Recompensa inmediata

Estrategia  $\epsilon$ -greedy: Equilibra exploración (probabilidad  $\epsilon$ ) y explotación ( $1-\epsilon$ )

Objetivo: Encontrar la política  $\pi^*$  que maximice la recompensa acumulada

La exploración se caracteriza por la búsqueda y la experimentación con nuevas alternativas, mientras que la explotación conlleva actividades relacionadas con el refinamiento y la aplicación de las capacidades, conocimientos o recursos existentes.

La métrica de éxito absoluto, es mantener la glucosa dentro del rango 70-180

class AgenteQLearning:

```
def __init__(self, n_estados=240, n_acciones=4):
    self.q_table = np.random.uniform(-1, 1, (n_estados, n_acciones)) * 0.1
        #Matriz de 240x4 multiplicada por 0,1
    self.alpha = 0.3      # Tasa de aprendizaje - # Óptimo para cambios graduales
    self.gamma = 0.95    # Factor de descuento - # considera acción en 4-6 horas
    self.epsilon = 0.3    # Exploración inicial      - # Óptimo para cambios graduales
```

- TAMAÑO Q-TABLE: 240 ESTADOS × 4 ACCIONES
- Descomposición de estados ( $4 \times 5 \times 4 \times 3 = 240$ ):
- Glucosa 4 niveles Rango crítico:
  - hipo(<70), objetivo(70-180), hiper(180-250), severa(>250)
- Insulina activa 5 niveles
  - 0, 5, 10, 15, 20U - cubre dosis comunes y efectos acumulativos
- Tiempo desde dosis 4 niveles Farmacocinética:
  - pico(0-60min), acción(60-120), decaimiento(120-240), residual(>240)
- Sensibilidad 3 niveles Variabilidad individual:
  - baja(resistente), normal, alta(sensible)

## BUCLE PRINCIPAL (1000 episodios)

Por cada episodio:

1. RESET → Estado inicial aleatorio (ej: glucosa 150 mg/dL)

2. BUCLE DE PASOS (hasta 24 horas simulado)

— a) DECIDIR DOSIS:

- $\epsilon$ -greedy:  $\epsilon$  decrece 30%→1%
- Si random <  $\epsilon$ : Acción aleatoria (0,5,10,15, 20U)
- Si no: Elegir acción con mayor Q-value en Q-table

— b) EJECUTAR EN SIMULADOR:

- Administrar insulina
- Simular 30 min de metabolismo
- Calcular nueva glucosa
- Calcular recompensa:
  - Glucosa 70-180: +5 (BUENO)
  - <70: -10 (HIPOGLUCEMIA)
  - >180: -2 a -5 (HIPERGLUCEMIA)

— c) ACTUALIZAR Q-TABLE (ecuación Bellman):

- $Q(s,a) \leftarrow Q(s,a) + 0.3 \times [R + 0.95 \times \max(Q(s')) - Q(s,a)]$
- Aprende si la acción fue buena/mala

— d) REPETIR hasta terminar episodio

# Descripción de los datos

Se creó una base de datos con 50.000 datos conteniendo las características descritas en el siguiente título, la distribución por genero es de 25.000/25.000

## Características seleccionadas

Característica	Descripción	Rango	Promedio	Desviación estandar
HbA1c	Análisis sanguíneo que refleja el control glucémico promedio de 2–3 meses	7.0–7.8%	~7.4%	~0.2
Glucosa en ayunas	Nivel de azúcar en sangre después de 8+ horas sin comer	100–160 mg/dL	~130 mg/dL	~15 mg/dL
Glucosa posprandial	Nivel de azúcar 2 horas después de comer	1.4–1.8× ayunas	~1.6× ayunas	~0.1×
Raciones de carbohidratos	Cantidad diaria de carbohidratos consumidos en raciones	65–80 raciones ±20%	~72 raciones	~20% del valor base
Ratio insulina/carbohidrato	Unidades de insulina necesarias por gramo de carbohidrato	500 / dosis_total	— (depende de dosis total)	Variable
Dosis basal	Insulina de acción prolongada para mantener niveles entre comidas	40–60% de la dosis total	~50%	~10%
Dosis bolo	Insulina rápida para comidas y correcciones	40–60% de la dosis total	~50%	~10%
Factor de sensibilidad	Reducción de glucosa (mg/dL) por unidad de insulina rápida	1800 / dosis_total	— (depende de dosis total)	Variable

Dosis total de insulina	Cantidad diaria total de insulina requerida	0.6–0.8 U/kg × peso	~0.7 U/kg × peso	~0.1 U/kg
-------------------------	---	---------------------	------------------	-----------

## Distribución Hombres

Edad	Altura promedio (cm)	Desv. altura (cm)	Peso promedio (kg)	Desv. peso (kg)
1	76.0	3.0	9.5	1.2
2	87.0	3.5	12.5	1.5
3	95.0	4.0	14.5	1.8
4	102.0	4.5	16.5	2.0
5	109.0	5.0	18.5	2.5
6	116.0	5.5	21.0	3.0
7	122.0	6.0	23.5	3.5
8	128.0	6.5	26.5	4.0
9	133.0	7.0	29.5	4.5
10	138.0	7.5	32.5	5.0
11	143.0	8.0	36.0	6.0
12	149.0	8.5	40.0	7.0
13	156.0	9.0	45.0	8.0
14	163.0	9.0	51.0	9.0
15	169.0	9.0	57.0	10.0
16	173.0	8.5	62.0	10.0
17	175.0	8.0	66.0	10.0
18	176.0	7.5	68.0	10.0
20	177.0	7.0	73.0	11.0
25	177.0	7.0	78.0	12.0
30	177.0	7.0	82.0	13.0
40	176.0	7.0	85.0	14.0
50	175.0	7.0	86.0	14.0
60	174.0	7.0	85.0	13.0
70	172.0	7.0	82.0	12.0

80	170.0	7.0	79.0	11.0
85	169.0	7.0	77.0	10.0

### Distribución Mujeres

Edad	Altura promedio (cm)	Desv. altura (cm)	Peso promedio (kg)	Desv. peso (kg)
1	74.0	3.0	9.0	1.1
2	86.0	3.5	12.0	1.4
3	94.0	4.0	14.0	1.7
4	101.0	4.5	16.0	1.9
5	108.0	5.0	18.0	2.3
6	115.0	5.5	20.5	2.8
7	121.0	6.0	23.0	3.3
8	127.0	6.5	26.0	4.0
9	132.0	7.0	29.0	4.5
10	138.0	7.5	32.5	5.0
11	144.0	8.0	36.5	6.0
12	150.0	8.5	41.0	7.0
13	156.0	8.5	46.0	8.0
14	160.0	8.0	50.0	8.0
15	162.0	7.5	53.0	8.0
16	163.0	7.0	55.0	8.0
17	164.0	7.0	56.0	8.0
18	164.0	7.0	57.0	8.0
20	164.0	7.0	60.0	9.0
25	164.0	7.0	63.0	10.0
30	164.0	7.0	66.0	11.0
40	163.0	7.0	68.0	12.0
50	162.0	7.0	69.0	12.0
60	161.0	7.0	68.0	11.0

70	159.0	7.0	66.0	10.0
80	157.0	7.0	64.0	9.0
85	156.0	7.0	62.0	9.0

Característica	Descripción	Rango / Categorías
Nivel de actividad física	Evaluación general del nivel de actividad	1–5 (1 = muy baja, 5 = muy alta)
Horas de actividad física	Horas semanales dedicadas a ejercicio o movimiento estructurado	Variable (dependiente del nivel y edad)
Tipo de ejercicio	Clasificación según intensidad	0 = sedentario, 1 = baja, 2 = moderada, 3 = alta intensidad

## Descripción función de recompensa

### Recompensas por nivel de glucosa

Rango de glucosa (mg/dL)	Categoría clínica	Recompensa	Justificación médica
< 54	Hipoglucemia severa	-20	EMERGENCIA — Riesgo de coma, convulsiones
54 – <70	Hipoglucemia	-10	PELIGROSO — Síntomas, requiere acción inmediata
70 – 180	Rango objetivo	+5	ÓPTIMO — Meta ADA, menor riesgo de complicaciones
180 – 250	Hiperglucemia leve	-2	SUBÓPTIMO — Riesgo aumentado a largo plazo
250 – 300	Hiperglucemia moderada	-5	MALO — Síntomas, riesgo de cetosis
> 300	Hiperglucemia severa	-10	PELIGROSO — Riesgo de cetoacidosis y deshidratación

### Recompensas adicionales (Bonus / Penalizaciones)

Situación	Recompensa	Condición	Propósito
Estabilidad glucémica	+1	Cambio de glucosa < 30 mg/dL en 1 hora	Premiar control estable
Corrección efectiva	+3	Glucosa sale de un rango peligroso	Premiar acción correctiva oportuna
Sobre-corrección	-5	Hipoglucemia después de un bolo grande	Penalizar dosis excesivas
Hipoglucemia inminente	-3	Glucosa < 80 y bajando rápido (> 2 mg/dL/min)	Premiar prevención
Tiempo en rango extendido	+0.1 por minuto	Cada minuto en 70–180 mg/dL	Premiar mantenimiento

**se supone que es una nota al pie**

cetosis: estado metabólico natural donde el cuerpo, ante la falta de carbohidratos (glucosa), empieza a quemar grasa para obtener energía

Cetoacidosis: complicación grave y potencialmente mortal de la diabetes que ocurre cuando el cuerpo, por falta de insulina, no puede usar la glucosa para energía y empieza a quemar grasa, produciendo cetonas que vuelven la sangre ácida

# Instrucciones de uso

El funcionamiento de los códigos descritos posteriormente no es automático una vez desencadenado el primer paso, sino que cada uno debe ejecutarse una vez concluida la ejecución del paso anterior:

- 1) generador\_pacientes.py: Es el encargado de crear la base de datos con las características descritas anteriormente en ‘Características seleccionadas’, en donde se crearán 50.000 pacientes dentro de las variaciones descritas previamente
- 2) entrenamiento\_checkpoints.py: Una vez creada la base de datos se establecen 100 modelos de Q Learning y compiten con 500 pacientes cada uno, el que mejor se desempeñe será coronado el campeón y encargado de proveer un plan para la administración de insulina de las 50.000 personas de la base de datos
- 3) test\_modelo\_final.py: Cuando se haya terminado de escoger el mejor modelo ‘best\_model’ como es referenciado dentro de los archivos, procederá a “administrar insulina” a los 50.000 pacientes
- 4) Una vez ejecutados estos códigos, se tiene el mejor modelo que puede en más del 90% del tiempo recomendar un plan que permita mantener la glucosa dentro del rango objetivo 70-180
  - a) Se tiene una opción aparte que permite simular el plan de administración para un paciente en específico

## Resultados

Al implementar el paso 1 descrito en el ítem anterior, se obtiene algo parecido a la figura 1; se puede observar directamente la base de datos yendo a la dirección ‘Base de datos/db\_diabetes\_50k.csv’

figura1.png

Al implementar el paso 2, se obtiene el mejor modelo en la dirección ‘Resultados/best\_model/\*’ teniendo múltiples maneras de ver el archivo (formato .csv, .json, .pkl, .nyp) y aparecerá un gráfico como el de la figura 2

figura2.png

Al implementar el paso 3, el código buscara el modelo en el ítem antes descrito y procederá a ‘administrar’ insulina para llevar a los pacientes al rango de 70-180, quedando con un gráfico como el de la figura 3

figura3.png

Si quiso aplicar la opción para un plan personalizado para un paciente específico, podrá observar gráficos como los de la figura4

figura4.png