



# Tecnológico de Monterrey

## **Proyecto 1:**

### **Análisis de datos para aplicaciones médicas**

**Instituto Tecnológico y de Estudios Superiores de Monterrey**  
Campus Guadalajara

Marissa Edith Luna Landa A01645653

Ximena Cantón Ayllón A01645105

### **Modelación del aprendizaje con Inteligencia Artificial**

Omar Mendoza Montoya

Grupo 302

**Fecha de entrega:** Viernes, 06 de junio de 2025.

## 1. Descripción de la aplicación implementada junto con los datos recolectados.

### ¿En qué consiste la aplicación?

La aplicación desarrollada tiene como objetivo clasificar en tiempo real una de seis actividades físicas distintas que realiza un usuario, utilizando datos del acelerómetro de un dispositivo móvil. Las actividades reconocidas por el modelo son:

- Nothing (reposo)
- Jumping (saltos)
- Lateral lunges (desplantes laterales)
- Hip circles (círculos de cadera)
- Walking straight (caminar en línea recta)
- Squats (sentadillas)

El sistema fue entrenado previamente con datos recolectados manualmente por miembros del equipo, realizando cada actividad durante intervalos definidos. Posteriormente, el modelo se integró en una aplicación que recibe nuevos datos de entrada y predice en tiempo real cuál de las actividades está ocurriendo.

### ¿Qué se espera que haga?

Se espera que la aplicación reciba señales del acelerómetro del dispositivo (en este caso, desde un celular con la app Phyphox). Una vez recibidas las señales, el modelo extrae las mejores características (features) de las observaciones, y con los mejores parámetros clasifica la actividad en tiempo real usando el modelo de Random Forest entrenado. Esperando que dicha predicción sea lo más precisa posible.

En esencia, la aplicación permite que un sistema reconozca automáticamente el tipo de movimiento físico que está realizando un usuario, sin intervención manual.

### ¿Qué ejemplos de aplicaciones hay afuera que hagan algo similar?

El reconocimiento automático de actividades humanas (Human Activity Recognition, HAR) es un área ampliamente estudiada que ya tiene aplicaciones prácticas en sectores como el fitness, la salud y la asistencia personalizada.

Por ejemplo, plataformas como **Google Fit**, **Apple Health** y dispositivos como **Fitbit**, **Garmin** o **Polar** emplean sensores inerciales (acelerómetros y giroscopios) para identificar si una persona está caminando, corriendo, en bicicleta o en reposo. Estas herramientas no solo reconocen actividades, sino que también permiten realizar un seguimiento continuo del estado físico, estimar calorías, registrar tiempo activo y brindar retroalimentación al usuario (Machorro-Cano et al., 2023).

Además, en el ámbito médico, los sistemas HAR han sido utilizados para apoyar procesos de **rehabilitación física**. Existen aplicaciones que permiten monitorear si un paciente realiza correctamente los ejercicios prescritos, incluso de forma remota. Esto ha sido especialmente útil en pacientes con movilidad limitada, como quienes se recuperan de accidentes cerebrovasculares (Chen et al., 2016). La incorporación de sensores portátiles facilita un seguimiento objetivo y automático de su progreso.

También se han desarrollado marcos de HAR integrados con **inteligencia artificial**, capaces de combinar datos fisiológicos, ambientales e inerciales para aumentar la precisión y adaptabilidad del sistema en distintos entornos (Demrozi et al., 2020). Esto abre la puerta a su uso en hogares inteligentes, monitoreo de adultos mayores y contextos de asistencia autónoma (Kaur & Kaur, 2024; Liu et al., 2021).

Nuestro proyecto sigue una línea similar: diseñamos una aplicación capaz de reconocer actividades físicas específicas a partir de señales del acelerómetro, entrenando modelos que puedan ejecutarse en línea y adaptarse a distintas personas. Este tipo de sistema tiene potencial en entornos deportivos, de rehabilitación o de monitoreo remoto, ya que permite detectar en tiempo real si se están realizando correctamente ciertos movimientos.

### **¿Qué tipo de información va a monitorear?**

La aplicación está diseñada para monitorear en tiempo real el movimiento corporal mediante el uso del acelerómetro de un dispositivo móvil. Específicamente, se registran los valores de aceleración en los tres ejes espaciales: **X (izquierda–derecha)**, **Y (adelante–atrás)** y **Z (arriba–abajo)**. A partir de estas señales, se extraen diversas características estadísticas que resumen el comportamiento del movimiento en periodos breves.

Para estructurar la recolección de datos, se definieron 10 ensayos (trials) por cada actividad, y dentro de cada ensayo se generaron 10 ventanas de análisis de 0.5 segundos cada una, con una tasa de muestreo de 30 Hz. Esto dio lugar a 100 observaciones por actividad, y considerando que se registraron 6 actividades diferentes en cada trial, se obtuvieron 600 observaciones por participante.

En cada ventana, se calcularon múltiples características, como media, desviación estándar, curtosis, asimetría, máximos, mínimos, rango peak-to-peak, derivadas, entre otras. Estas características conforman un vector de entrada que se utiliza para alimentar los modelos de clasificación.

El objetivo final de la aplicación es reconocer automáticamente qué actividad física está realizando la persona en función de los patrones de aceleración capturados, lo cual permite aplicaciones tanto en el ámbito deportivo como clínico.

## **2. Descripción de las características extraídas de los datos.**

A partir de los datos originales, se obtuvieron otras métricas que se anexaron como características en la base de datos. Esto se hace debido a que las señales crudas son más

difíciles de interpretar en comparación a las características estadísticas, lo que facilita la comprensión de cómo y por qué un modelo toma ciertas decisiones. (Fulcher & Jones, 2014).

El objetivo de estas características es capturar distintos aspectos del movimiento, como su intensidad, variabilidad, simetría o brusquedad, para que el modelo pueda distinguir entre actividades similares.

Las características extraídas por eje (X, Y, Z) fueron:

- **Media (mean):** promedio de la aceleración en la ventana.
- **Desviación estándar (std):** variabilidad de la señal.
- **Curtosis (kurtosis):** presencia de valores extremos.
- **Asimetría (skewness):** asimetría o sesgo de la señal.
- **Mínimo y máximo:** valores extremos de aceleración.
- **Mediana:** valor central de la distribución.
- **Percentiles 25 y 75:** valores que describen la dispersión intercuartílica.
- **Peak-to-peak (ptp):** diferencia entre el valor máximo y mínimo.
- **Media y desviación de la derivada:** estimación del cambio de aceleración (suavidad o brusquedad del movimiento).

Adicionalmente, se incluyeron características combinadas como la **magnitud total del vector de aceleración**, calculada como:

$$\sqrt{X^2 + Y^2 + Z^2}$$

En total, se extrajeron **37 características por ventana**, lo que permitió construir vectores de entrada representativos para el modelo de aprendizaje automático.

### **3. Resultados obtenidos en la evaluación de los clasificadores probados, incluyendo la obtención de hiperparámetros y selección de características.**

En total se probaron diez clasificadores distintos. A continuación, se presenta una ficha resumen para cada modelo con los resultados obtenidos.

#### **Accuracy, precision y recall:**

- SVM lineal

Accuracy (Exactitud): 0.89

| Clase | Precisión | Recall (Sensibilidad) |
|-------|-----------|-----------------------|
| 1     | 0.93      | 0.93                  |
| 2     | 0.85      | 0.89                  |
| 3     | 0.88      | 0.82                  |

|   |      |      |
|---|------|------|
| 4 | 0.85 | 0.87 |
| 5 | 0.96 | 0.98 |
| 6 | 0.88 | 0.86 |

- SVM RBF

Accuracy (Exactitud):

| Clase | Precisión | Recall (Sensibilidad) |
|-------|-----------|-----------------------|
| 1     | 1.00      | 0.94                  |
| 2     | 0.86      | 0.83                  |
| 3     | 0.95      | 0.76                  |
| 4     | 0.72      | 0.90                  |
| 5     | 0.91      | 0.98                  |
| 6     | 0.83      | 0.78                  |

- LDA

Accuracy (Exactitud): 0.83

| Clase | Precisión | Recall (Sensibilidad) |
|-------|-----------|-----------------------|
| 1     | 1.00      | 0.93                  |
| 2     | 0.84      | 0.71                  |
| 3     | 0.97      | 0.72                  |
| 4     | 0.70      | 0.85                  |
| 5     | 0.86      | 0.98                  |
| 6     | 0.71      | 0.78                  |

- K-NN

Accuracy (Exactitud): 0.91

| Clase | Precisión | Recall (Sensibilidad) |
|-------|-----------|-----------------------|
| 1     | 1.00      | 0.96                  |
| 2     | 0.96      | 0.93                  |

|   |      |      |
|---|------|------|
| 3 | 0.98 | 0.84 |
| 4 | 0.79 | 0.94 |
| 5 | 0.94 | 0.97 |
| 6 | 0.88 | 0.86 |

- MLP de dos capas

Accuracy (Exactitud): 0.91

| Clase | Precisión | Recall (Sensibilidad) |
|-------|-----------|-----------------------|
| 1     | 0.97      | 0.95                  |
| 2     | 0.94      | 0.94                  |
| 3     | 0.94      | 0.88                  |
| 4     | 0.85      | 0.90                  |
| 5     | 0.94      | 0.96                  |
| 6     | 0.88      | 0.87                  |

- Random Forest

Accuracy (Exactitud): 0.93

| Clase | Precisión | Recall (Sensibilidad) |
|-------|-----------|-----------------------|
| 1     | 0.99      | 0.96                  |
| 2     | 0.96      | 0.95                  |
| 3     | 0.97      | 0.90                  |
| 4     | 0.85      | .91                   |
| 5     | 0.94      | 0.97                  |
| 6     | 0.90      | 0.89                  |

- Naive Bayes

Accuracy (Exactitud): 0.89

| Clase | Precisión | Recall (Sensibilidad) |
|-------|-----------|-----------------------|
| 1     | 0.99      | 0.94                  |

|   |      |      |
|---|------|------|
| 2 | 0.93 | 0.91 |
| 3 | 0.91 | 0.79 |
| 4 | 0.81 | 0.89 |
| 5 | 0.95 | 0.97 |
| 6 | 0.80 | 0.86 |

- Extra trees

Accuracy (Exactitud): 0.82

| Clase | Precisión | Recall (Sensibilidad) |
|-------|-----------|-----------------------|
| 1     | 0.94      | 0.91                  |
| 2     | 0.81      | 0.79                  |
| 3     | 0.76      | 0.80                  |
| 4     | 0.74      | 0.79                  |
| 5     | 0.95      | 0.92                  |
| 6     | 0.75      | 0.72                  |

- Gradient Boosting

Accuracy (Exactitud): 0.92

| Clase | Precisión | Recall (Sensibilidad) |
|-------|-----------|-----------------------|
| 1     | 0.99      | 0.94                  |
| 2     | 0.93      | 0.95                  |
| 3     | 0.96      | 0.90                  |
| 4     | 0.84      | 0.90                  |
| 5     | 0.94      | 0.96                  |
| 6     | 0.92      | 0.90                  |

- Ridge Classifier

Accuracy (Exactitud): 0.77

| Clase | Precisión | Recall (Sensibilidad) |
|-------|-----------|-----------------------|
|-------|-----------|-----------------------|

|   |      |      |
|---|------|------|
| 1 | 1.00 | 0.90 |
| 2 | 0.79 | 0.70 |
| 3 | 0.90 | 0.77 |
| 4 | .80  | 0.59 |
| 5 | 0.65 | 0.99 |
| 6 | 0.69 | 0.72 |

### Obtención de hiperparámetros y características:

#### - SVM lineal

**Hiperparámetro optimizado:** coeficiente de regularización (  $C$  ).

El valor óptimo de  $C$  se encontró utilizando `GridSearchCV()`. En la siguiente gráfica podemos ver que conforme  $C$  aumenta, también lo hace la exactitud del modelo, hasta llegar un momento en el que estabiliza y la exactitud se mantiene en rangos muy similares.

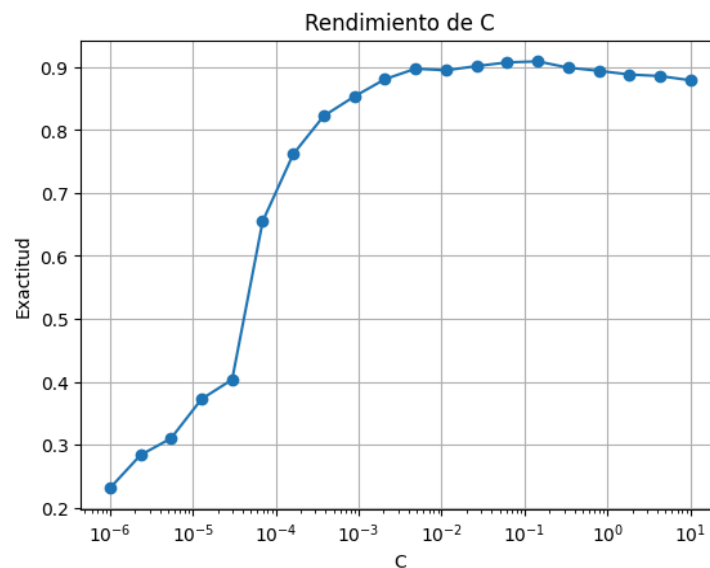


Fig 1. Rendimiento de  $C$  para modelo SVM lineal

**Características utilizadas:** de las 37 características, se realizó validación cruzada con para evaluar distintas selecciones de características obtenidas con **Recursive Feature Elimination** `RFE()`.



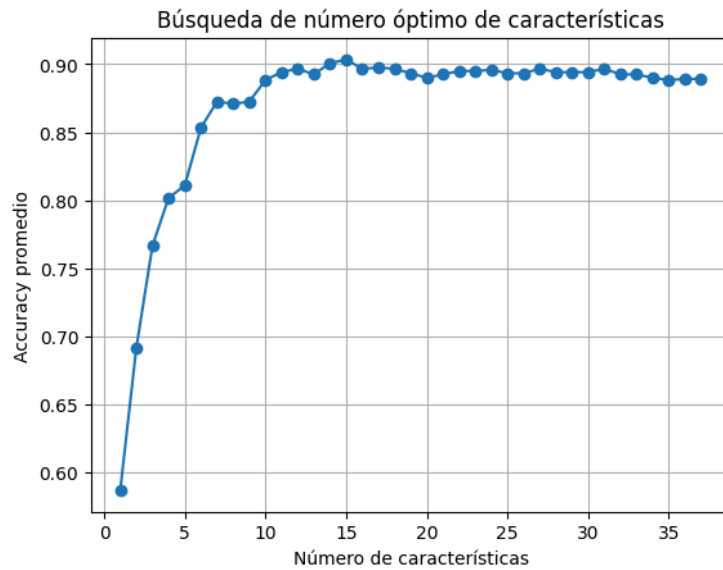


Fig 2. Número óptimo de característica para modelo SVM lineal

### - Random Forest

**Hiperparámetros utilizados:** número de árboles ( $n\_estimators$ ).

Se graficó el rendimiento en exactitud del hiperparámetro  $n\_estimators$ . En la siguiente ilustración se observa como en el rango considerado de árboles, la exactitud se mantiene alta (entre 0.90 y 0.94).

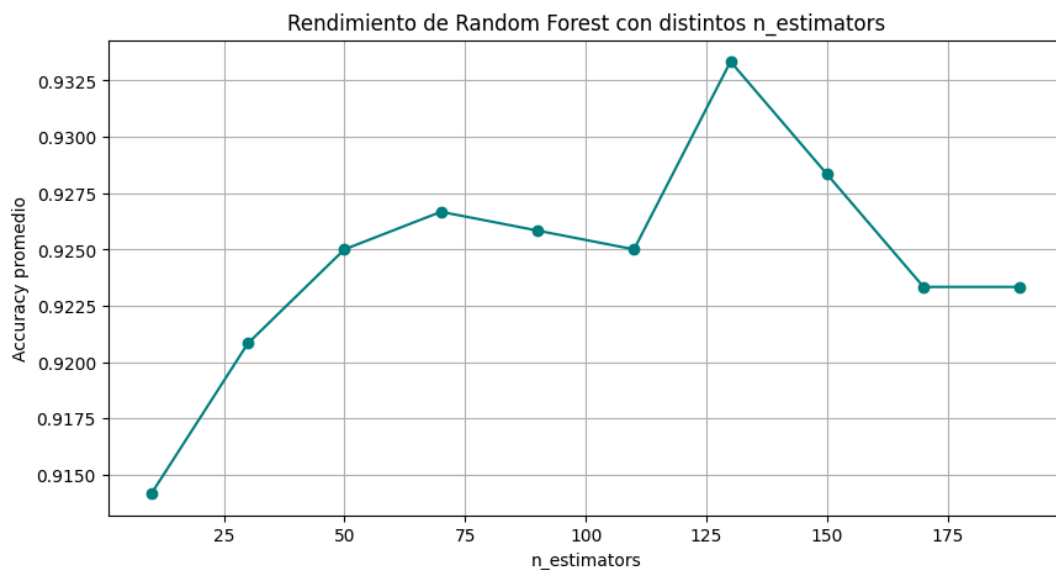


Fig 3. Rendimiento de  $n\_estimators$  para modelo Random Forest

**Características utilizadas:** Nuevamente se utilizó RFE para la búsqueda de características, obteniendo que a partir de 18 características, la exactitud se estabiliza.

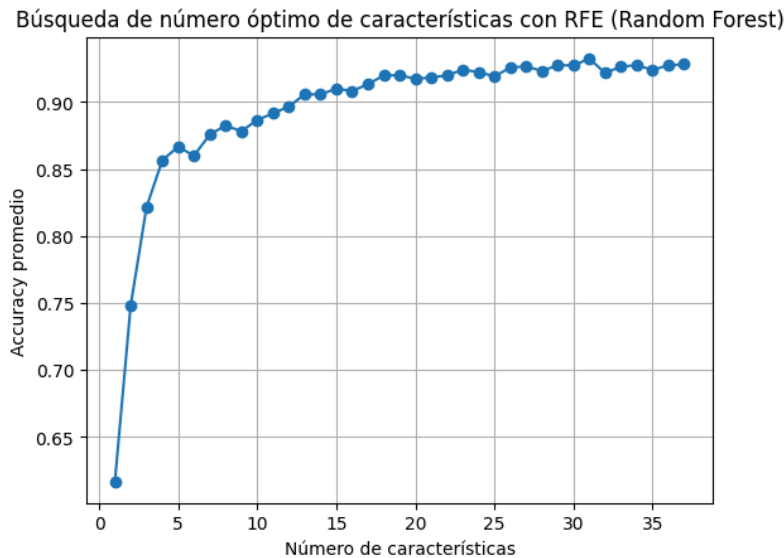


Fig 4. Número óptimo de característica para modelo Random Forest

#### 4. Resultados de la aplicación en línea y respuesta a las siguientes preguntas:

##### ¿Funciona igual con todos los miembros del equipo?

Inicialmente se entrenaron modelos por separado para cada persona del equipo, y se observó que el rendimiento de la clasificación en línea variaba notablemente según la persona que realizaba la actividad. Esto sugiere que el modelo aprendía patrones específicos del movimiento de cada individuo (como intensidad o ritmo).

Posteriormente, se optó por unificar los datos en una sola base de entrenamiento, con registros de todos los integrantes. Aunque esto provocó una ligera disminución en la exactitud promedio, se logró una mayor generalización: el modelo funcionó de forma más equilibrada para todos los miembros del equipo, sin depender de quién realizaba la actividad.

##### ¿El rendimiento de la aplicación en línea es el esperado de acuerdo a los resultados de validación cruzada?

Sí. El rendimiento observado en la aplicación en línea fue coherente con los resultados obtenidos durante la validación cruzada. Las actividades fueron reconocidas con precisión en tiempo real, especialmente aquellas con patrones más definidos como *jumping* o *walking straight*. Por otra parte, actividades como *hip circles* sí fue de las más difíciles de predecir en la práctica, lo cual coincidió con que en la validación cruzada éste ejercicio fue el que tuvo menor precisión y sensibilidad.

Las diferencias menores de rendimiento pueden deberse a variaciones en la ejecución de las actividades o ruido en la señal, pero en general, el modelo respondió de forma confiable y estable.

## 5. Conclusiones

### Marissa:

Este proyecto me ayudó a ver los modelos de machine learning desde otra perspectiva. Es muy sencillo entrenar un modelo y ya, pero ahora entiendo que es mucho más que eso. Es un sistema que hay que entender, evaluar y ajustar, y todo ese proceso tiene un impacto directo en qué tan bien funciona.

Una de las cosas con las que más me quedo, es la importancia de la validación cruzada. Esta técnica no solo sirve para tener una idea del rendimiento general, sino también para evaluar por clase. Un modelo puede tener una buena exactitud global, pero estar fallando en clases específicas, lo cual sería un problema si esas clases son relevantes en la aplicación real.

Otra parte que me pareció clave fue darme cuenta de que ajustar hiper parámetros y seleccionar características no son tareas mecánicas, sino decisiones que deben tomarse con base en una comprensión real del modelo. Entender cómo afectan parámetros como `C` en SVM o `n_estimators` en Random Forest te permite optimizar de forma más consciente, en lugar de solo probar combinaciones al azar. Además, aprender sobre selección de características fue una revelación: me mostró cómo reducir la dimensionalidad del problema, enfocarme en lo realmente relevante, ahorrar tiempo computacional y, al mismo tiempo, mejorar la interpretación y eficiencia del modelo en general.

En general, este proyecto me dio una visión más completa del ciclo de vida de un modelo: desde la adquisición y procesamiento de datos hasta su implementación en una aplicación funcional en línea. Me quedo con muchas herramientas útiles, pero sobre todo con el aprendizaje de que no se trata solo de usar modelos, sino de saber cómo y por qué funcionan, para poder mejorarlos de forma real.

### Ximena:

Este proyecto me permitió comprender el desarrollo de modelos de machine learning desde una perspectiva mucho más completa. Más allá de simplemente entrenar un modelo, pude ver que todo el sistema —desde la adquisición de datos, el procesamiento, la validación, hasta la implementación en línea— tiene un impacto directo en el rendimiento final.

Algo que realmente me marcó fue darme cuenta de que un modelo puede tener métricas aceptables en general, pero aún así fallar en las clases más importantes. Esto me ayudó a valorar el papel de la validación cruzada no solo como una forma de medir precisión global, sino también como una herramienta clave para entender errores específicos y tomar decisiones más informadas.

Además, experimenté de forma directa lo importante que es entender los hiper parámetros y la selección de características. No se trata simplemente de buscar la mejor combinación, sino

de conocer cómo y por qué esos elementos influyen en el comportamiento del modelo. Ese tipo de comprensión me dio mucha más claridad al momento de ajustar y optimizar.

Finalmente, llevar el modelo a una aplicación funcional en línea me hizo conectar todo el proceso y valorar cada etapa. Me llevo una visión más amplia y práctica de lo que implica diseñar un sistema inteligente: no basta con que el modelo “funcione”, también es importante entender su lógica, sus límites y cómo mejorarlo.

## 6. Referencias

Chen, K., Li, Y., & Zhou, W. (2016). *Evaluation of a smartphone human activity recognition application with able-bodied and stroke participants*. *Journal of NeuroEngineering and Rehabilitation*, 13(1), 1–10. <https://jneuroengrehab.biomedcentral.com/articles/10.1186/s12984-016-0114-0>

Demrozi, F., Pravadelli, G., Bihorac, A., & Rashidi, P. (2020). *Human Activity Recognition using Inertial, Physiological and Environmental Sensors: A Comprehensive Survey*. arXiv preprint arXiv:2004.08821. <https://ieeexplore.ieee.org/document/9257355>

Fulcher, B. D., & Jones, N. S. (2014). Highly comparative feature-based time-series classification. *IEEE Transactions on Knowledge and Data Engineering*, 26(12), 3026–3037. <https://doi.org/10.1109/TKDE.2014.2316504>

Kaur, P., & Kaur, P. (2024). *Human activity recognition: A comprehensive review*. *Expert Systems*, 41(1), e13680. <https://onlinelibrary.wiley.com/doi/10.1111/exsy.13680>

Liu, R., Ramli, A. A., Zhang, H., Henricson, E., & Liu, X. (2021). *An overview of human activity recognition using wearable sensors: Healthcare and artificial intelligence*. arXiv preprint arXiv:2103.15990. [https://link.springer.com/chapter/10.1007/978-3-030-96068-1\\_1](https://link.springer.com/chapter/10.1007/978-3-030-96068-1_1)

Machorro-Cano, I., Olmedo-Aguirre, J. O., Alor-Hernández, G., Rodríguez-Mazahua, L., Sánchez-Morales, L. N., & Pérez-Castro, N. (2023). *Cloud-based platforms for health monitoring: A review*. *Informatics*, 10(1), 1–20. <https://www.mdpi.com/2227-9709/10/1/3>