INFORME FINAL
SISTEMA DE CLASIFICACIÓN DE ACTIVIDADES HUMANAS

Vanessa Sánchez Morales
David Artunduaga Penagos
Juan Jose de la Pava
Daniel José Plazas Cortés

Universidad ICESI
Facultad Barberi de Ingeniería, Diseño y Ciencias Aplicadas
Inteligencia Artificial I
Milton Orlando Sarria Paja

Cali, Colombia
22 de Noviembre de 2025

Presentación

El presente informe tiene como objetivo documentar de manera técnica y estructurada el desarrollo de un sistema de clasificación automática de acciones humanas a partir de secuencias de video procesadas mediante técnicas de visión por computador y algoritmos de aprendizaje automático.

Este informe explica detalladamente cada módulo, la metodología, los parámetros utilizados, el funcionamiento interno de los scripts y los resultados obtenidos.

Introducción

La clasificación automática de acciones es un área clave dentro de la visión por computador debido a sus aplicaciones en salud, deporte, seguridad, ergonomía industrial y análisis biomecánico. El presente proyecto implementa un pipeline completo para reconocer acciones a partir de la información anatómica obtenida de una persona en movimiento.

Se emplean técnicas modernas como:

- Extracción de *landmarks* tridimensionales.
- Normalización espacial del cuerpo. Cálculo de ángulos articulares. Características estadísticas en ventanas temporales.
- Entrenamiento con modelos supervisados y optimización de los mismos.

El objetivo final es construir un sistema robusto, interpretable y capaz de diferenciar diversas acciones.

Metodología

El proceso general se estructura en 3 fases, de acuerdo a los lineamientos y las buenas prácticas en la construcción de sistemas de aprendizaje automático:

Fase 1 - Extracción de Landmarks

La fase inicial del sistema consiste en la **extracción de landmarks corporales** a partir de los videos crudos. Este procedimiento se realiza *frame por frame* y utiliza la librería **MediaPipe Pose**, la cual permite obtener una representación anatómica del cuerpo humano mediante 33 puntos clave. Cada punto incluye coordenadas tridimensionales (x, y, z) y un valor adicional denominado *visibility*, que indica la fiabilidad de la detección en ese frame.

El código implementa esta fase siguiendo los pasos que se describen a continuación:

1. Lectura del video y descomposición en dataframes:
    - Se utiliza cv2.VideoCapture para abrir cada video registrado en el dataset.
    - El código recorre el video secuencialmente utilizando un bucle while, extrayendo cada frame mediante cap.read().
    - Cada frame se transforma de BGR a RGB para ser compatible con MediaPipe.

2. Detección de landmarks con MediaPipe Pose

    - Para cada frame, se aplica el modelo mp_pose.Pose(...), que retorna la estructura results, dentro de la cual se encuentran los 33 landmarks corporales si la detección es exitosa.
    - Cada landmark contiene:
        - x: coordenada horizontal normalizada (0–1).
        - y: coordenada vertical normalizada (0–1).
        - z: profundidad relativa con respecto al plano del cuerpo.
        - visibility: probabilidad (0–1) de que el punto esté correctamente detectado.

3. Conversión del resultado del modelo a estructura tabularLos landmarks detectados se recorren uno por uno mediante un bucle que itera sobre results.pose_landmarks.landmark.

    - Para cada punto anatómico, el sistema crea un registro que incluye:
        - Nombre del video de origen.
        - Acción asociada (etiqueta de clase).
        - Índice del frame.
        - Coordenadas x, y, z y visibilidad.

    - Esta información se almacena en un diccionario temporal y posteriormente se añade a un DataFrame global.

4. Estandarización, filtrado y control de calidad

    - Los valores de x, y y visibility permanecen normalizados entre 0 y 1, lo que permite comparar datos de diferentes resoluciones o cámaras.
    - En los casos en que la visibilidad de un landmark es muy baja, el código puede identificar el valor y permitir filtrados posteriores durante el preprocesamiento.
    - Si un frame no contiene detección válida, se registra como "frame omitido" o se coloca una fila con valores NaN (dependiendo del flujo definido).

5. Exportación de landmarks del video
    - Una vez procesado el video completo, los landmarks se guardan en un archivo .csv
    - Cada archivo contiene miles de filas (según la duración del video) y 33×4 = 132 columnas asociadas a los landmarks, además del nombre del video, acción y número de frame.

En conjunto, esta fase convierte los videos originales en una **representación matemática estructurada del movimiento humano**, que sirve como insumo directo para la extracción de características biomecánicas desarrollada en las fases posteriores. Gracias al uso de MediaPipe Pose, los landmarks extraídos mantienen estabilidad ante variaciones de iluminación, fondo y posición del sujeto, lo que permite disponer de datos consistentes para el análisis temporal y la clasificación.

Fase 2 - Ingeniería de Características

La segunda fase del sistema consiste en transformar los **landmarks corporales** obtenidos en la etapa de captura en **representaciones numéricas compactas y útiles** para que los modelos de machine

learning puedan aprender patrones de movimiento. Esta fase aplica una combinación de **normalización espacial, geometría vectorial, biomecánica y estadística multivariable**. A continuación, se describe el proceso:

1. Normalización de landmarks

   Los landmarks de MediaPipe vienen expresados en coordenadas (x, y, z) relativas a la imagen, pero con variaciones debidas a:

   - Distancia de la cámara
   - Altura de la persona
   - Posición en pantalla

   Para eliminar estas variaciones, el sistema los **normaliza** mediante dos pasos:

   a) Traslación al centro de la cadera

   Se calcula el punto medio entre ambas caderas:

$$C_{\text{cadera}} = \frac{C_{\text{izq}} + C_{\text{der}}}{2}$$

   Luego cada landmark se desplaza:

$$P_i' = P_i - C_{\text{cadera}}$$

   Esto centra el sistema de coordenadas en el tronco superior.

   b) Escalamiento por distancia entre hombros

   La distancia entre hombro izquierdo y derecho define un factor de escala:

$$d_{\text{hombros}} = \|H_{\text{izq}} - H_{\text{der}}\|$$

   Cada punto se divide por dicha distancia, lo que permite que los movimientos sean comparables entre personas de diferentes tallas y distancias a cámara.

2. Cálculo de ángulos biomecánicos

   Esta etapa extrae **8 ángulos articulares clave** del cuerpo: codos, hombros, caderas y rodillas.

   Cada ángulo se obtiene usando la definición del ángulo entre dos vectores:

   Para tres puntos p1,p2 y p3, el ángulo en p2 es:

$$\theta = \arccos\left(\frac{(p_1 - p_2)\cdot(p_3 - p_2)}{\|p_1 - p_2\|\,\|p_3 - p_2\|}\right)$$

   Esto refleja biomecánicamente:

   - Flexión-extensión del codo
   - Elevación/abducción del hombro
   - Flexión de cadera
   - Flexión de rodilla

3. Cálculo de inclinación lateral del tronco

   Para medir asimetrías posturales, se calcula el ángulo de la línea formada por ambos hombros:

$$v = H_{\text{der}} - H_{\text{izq}}$$

   La inclinación es:

$$\theta = |\arctan 2(v_y, v_x)|$$

Un tronco vertical tiene un ángulo cercano a 0°; inclinaciones laterales aumentan este valor.

4. Ventanas temporales de 15 frames

   Los ángulos no se analizan de forma aislada. Se usa una **ventana temporal de 15 frames**, equivalente a aproximadamente medio segundo. Lo anterior permite capturar patrones de movimiento, no solo poses.

   Se obtiene entonces una matriz de la siguiente forma:

$$W = \begin{bmatrix} a_{1,1} & \cdots & a_{1,8} \\ \vdots & \ddots & \vdots \\ a_{15,1} & \cdots & a_{15,8} \end{bmatrix}$$

5. Cálculo de velocidades angulares

   Esto captura cambios dinámicos como:

   - Aceleraciones del brazo
   - Intensidad del movimiento
   - Velocidad de ejecución.

   Dicha información se obtiene a partir del cálculo de la derivada discreta, con v1=0 para mantener dimensiones coherentes.

6. Características estadísticas (48 features totales)

   Sobre la ventana temporal se calculan estadísticas para resumir el comportamiento, el cual queda resumido en un vector de 48 componentes (48 features en total):

   a) Estadísticas de los ángulos (media, desviación estandar, mínimos y máximos)

   b) Estadísticas de velocidades (media de velocidades y desviación estandar)

7. Validación de Landmarks
   Para evitar ruido, se descartan frames donde los landmarks más críticos tengan visibilidad baja.

Fase 3 - Entrenamiento de modelos

La tercera fase del proyecto corresponde al entrenamiento de los modelos de aprendizaje supervisado encargados de clasificar automáticamente la actividad humana observada en cada ventana temporal del video.

En esta etapa se utilizan como entrada las características biomecánicas generadas en la Fase 2 (ángulos articulares, velocidades angulares y estadísticas por ventana), previamente normalizadas y agrupadas en un solo archivo estructurado.

El objetivo principal de esta fase es construir un clasificador robusto capaz de diferenciar acciones humanas a partir de señales cinemáticas, optimizando sus hiperparámetros de manera sistemática mediante **GridSearchCV** y aplicando buenas prácticas de división de datos, codificación y estandarización.

A continuación, se describe el pipeline de entrenamiento y se detallan los modelos utilizados.

1. Preparación del dataset para entrenamiento

   El proceso inicia cargando el archivo model_features.csv, resultado final de la ingeniería de características. Este archivo contiene:

   - 48 características biomecánicas por ventana temporal.

- Identificación del video (video_filename).
- Índice del frame central (frame_idx).
- Etiqueta de acción (action).

Los pasos principales de preparación son:

a) Separación de características (X) y etiquetas (y)
b) Codificación de etiquetas: Con LabelEncoder se se codifica y para poder ser procesado por el modelo
c) División en entrenamiento y prueba

2. Pipeline de Entrenamiento

Ambos modelos entrenados siguen un pipeline común que encapsula las siguientes etapas:

a) Estandarización: Se usa StandardScaler para igualar escalas entre características.
b) Reducción de dimensionalidad: Se aplica PCA con una retención del 95% de la varianza, reduciendo el ruido y mejorando la estabilidad del aprendizaje.
c) Entrenamiento del modelo: Se entrena uno de los dos modelos escogidos y se optimizan mediante **GridSearchCV**, probando múltiples combinaciones de parámetros.

Para el proyecto se utilizaron dos diferentes modelos, los cuales se describen a continuación:

1. Random Forest Classifier

Random Forest es un modelo de aprendizaje supervisado basado en un conjunto de árboles de decisión entrenados de forma paralela. Cada árbol aprende sobre una muestra aleatoria del dataset y sobre un subconjunto aleatorio de características, lo cual:

- Reduce el sobreajuste presente en modelos basados en árboles individuales.
- Aumenta la capacidad de generalización.
- Proporciona interpretabilidad en términos de importancia de características.

El clasificador final se obtiene mediante votación mayoritaria entre los árboles.

La detección de actividades humanas depende de patrones complejos en los ángulos y velocidades articulares, que no necesariamente siguen una distribución lineal, contienen interacciones entre características e incluyen ruido debido a variaciones en la captura.

Random Forest es especialmente apropiado porque:

- Maneja relaciones no lineales.
- Es robusto al ruido en las features.
- No requiere supuestos fuertes sobre la distribución de los datos.
- Funciona bien con un número moderado de características como el presente.
- Su estructura en árboles permite capturar decisiones jerárquicas, similares al análisis biomecánico humano.

Además, su eficiencia computacional lo hace ideal como primer modelo base para comparar con otros métodos más complejos.

2. XGBoost Classifier

   XGBoost (Extreme Gradient Boosting) es un algoritmo basado en **boosting de árboles**, donde cada árbol sucesivo corrige los errores cometidos por el árbol anterior. Sus ventajas comprenden su manejo avanzado del sesgo y la varianza, la regularización interna para evitar sobreajuste, su excelente rendimiento en tareas de clasificación estructurada y la optimización basada en gradientes.

   XGBoost es actualmente uno de los modelos más potentes para datos tabulares, especialmente cuando la estructura presenta correlaciones complejas entre variables.

   XGBoost es adecuado para el problema en mano porque:

   - Aprende relaciones no lineales y dependencias combinadas entre features.
   - Tolera muy bien datos con ruido y características correlacionadas.
   - Se beneficia del PCA previo reduciendo redundancias.
   - Mejora la capacidad discriminativa entre acciones similares (por ejemplo, flexión vs. extensión).

   Dado que los movimientos humanos tienen transiciones suaves pero diferenciables, XGBoost puede capturar estas variaciones con alta precisión.

Finalmente en esta etapa se guardan los modelos optimizados en archivos .joblib para ser utilizados en la aplicación en tiempo real.

Fase 4 - Aplicación en Tiempo Real

La cuarta y última fase del proyecto consiste en la construcción de un sistema operativo capaz de detectar en tiempo real la actividad humana realizada frente a una cámara. A diferencia de las fases anteriores —que operan de forma offline sobre datos previamente registrados— esta etapa integra captura de video, estimación de pose, extracción de características, inferencia del modelo y visualización interactiva en un solo pipeline de ejecución continua.

La implementación utiliza dos módulos principales:

- real_time_system.py: núcleo del sistema, encargado del procesamiento del video y la clasificación en tiempo real.

- utils.py: conjunto de funciones auxiliares reutilizables para cálculo geométrico, normalización, extracción de ángulos y otros componentes matemáticos.

A continuación se describe el funcionamiento completo del sistema y cada uno de sus componentes.

1. Flujo general del sistema

   En cada frame capturado por la cámara, el sistema ejecuta el siguiente pipeline:

   1. Captura del frame (OpenCV)
   2. Detección del cuerpo con MediaPipe Pose
   3. Normalización de landmarks
   4. Cálculo de los 8 ángulos biomecánicos

5. Construcción de una ventana temporal de 15 frames
6. Generación de 48 características (mismas usadas en entrenamiento)
7. Predicción del modelo Random Forest/XGBoost Suavizado temporal de la predicción
8. Visualización en pantalla de la actividad y nivel de confianza

Este ciclo se ejecuta alrededor de 25–30 veces por segundo.

2. Estimación de pose y extracción de características

   MediaPipe Pose proporciona 33 puntos tridimensionales por frame. Estos landmarks se trasladan al centro de la cadera, se normalizan por la distancia entre hombros y se procesan para obtener 8 ángulos articulares (codos, hombros, caderas y rodillas).

   El archivo utils.py contiene funciones para normalizar los landmarks, extraer los ángulos y calcular estadísticas sobre una ventana temporal.

   Cuando se acumulan 15 frames consecutivos, se generan 48 características que resumen la postura y dinámica del movimiento.

3. Clasificación en tiempo real

   Una vez obtenidas las características, el sistema:

   - Carga el modelo previamente entrenado y escogido (best_random_forest_model.joblib)
   - Predice la acción correspondiente y su probabilidad
   - Utiliza un historial de las últimas predicciones para suavizar cambios bruscos
   - Implementa un estado adicional de “quieto” basado en la baja variabilidad angular.

   Esto mejora la estabilidad y evita fluctuaciones rápidas en la interfaz.

4. Interfaz visual y experiencia del usuario

   El sistema genera una interfaz en OpenCV que incluye:

   - El esqueleto detectado
   - La acción predicha en tiempo real
   - Un indicador circular de confianza
   - Barras de nivel (bajo, medio, alto)
   - Estado del detector
   - Contador de frames

   Este diseño permite que el usuario observe la clasificación de forma clara y continua.

5. Comportamiento y rendimiento

   El sistema es eficiente gracias a:

   - MediaPipe en modo rápido
   - Cálculos vectorizados en NumPy
   - Uso de Random Forest para inferencia inmediata
   - Un contador de FPS incluido en utils.py.

   En caso de pérdida de detección o visibilidad baja, los búferes se reinician para evitar predicciones incorrectas.

En general, esta etapa convierte el modelo entrenado en una aplicación funcional capaz de reconocer actividades humanas directamente desde la cámara, integrando detección de pose, procesamiento biomecánico y clasificación temporal en un sistema estable y en tiempo real.

Con ello se completa el objetivo final del proyecto: un prototipo plenamente operativo para reconocimiento automático de acciones humanas.

Análisis de Resultados

A continuación se presenta la evaluación de los dos modelos de clasificación supervisada implementados, Random Forest y XGBoost, sobre el conjunto de datos de prueba. La evaluación se centra en métricas de rendimiento clave (Accuracy, Precision, Recall, F1-Score) y en el análisis detallado de la capacidad de cada modelo para discriminar entre las diferentes acciones humanas a través de sus respectivas Matrices de Confusión.

Random Forest

El modelo Random Forest muestra un rendimiento casi perfecto, con solo **15 errores** de clasificación sobre el total de predicciones (2227 muestras).

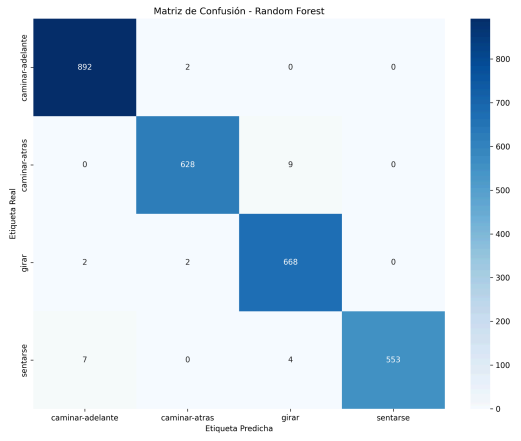

Figura 1. Matriz de confusión para Random Forest

Los errores se concentran principalmente en la confusión de la acción **"caminar-atras"** como **"girar"** (9 instancias) y **"sentarse"** como **"caminar-adelante"** (7 instancias). Esto puede deberse a la superposición en el movimiento de la cadera o a las primeras fases de la acción de sentarse, que contienen una desaceleración similar a una pausa antes del paso.

XGBoost

El modelo XGBoost también presenta un rendimiento sobresaliente, registrando un total de **20 errores** de clasificación sobre las 2227 muestras.

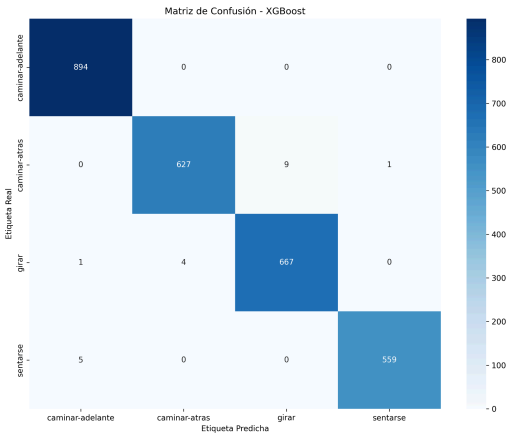

Figura 2. Matriz de confusión para XGBoost

A su vez, se realizó un gráfico de comparación para las métricas agregadas de ambos modelos. En este se observa que ambos modelos exhibieron un rendimiento excepcional, alcanzando una Accuracy (Exactitud), Precision, Recall y F1-Score perfectos de 1.0 .

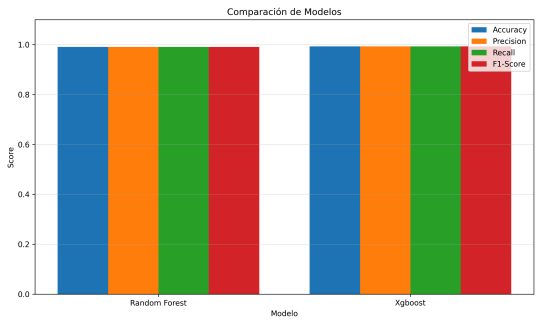

Figura 3. Comparación de modelos

Esta alta concordancia sugiere que las **48 características biomecánicas** extraídas (ángulos, velocidades y estadísticas temporales) son altamente discriminativas y capturan de forma robusta la cinemática esencial de las acciones clasificadas ("caminar-adelante", "caminar-atrás", "girar", "sentarse").

Como se observa, ambos modelos demuestran ser excelentes clasificadores con base en el conjunto de características diseñado en la Fase 2. Sin embargo, para la implementación final, se seleccionó el modelo **Random Forest**.

Este modelo de clasificación mostró un número total de errores ligeramente menor que XGBoost (15 vs 20), lo que, aunque marginal, indica una capacidad de generalización superior en este conjunto de datos. A su vez, Random Forest es conocido por su eficiencia computacional en la etapa de inferencia y su robustez ante el ruido en las *features*, características fundamentales para una aplicación en tiempo real.

Finalmente, la estructura de *ensemble* de árboles de decisión en Random Forest permite la interpretabilidad a través de la importancia de las características, una ventaja clave para comprender qué movimientos son decisivos en la clasificación.

Conclusiones

El proyecto culminó con el desarrollo y la implementación exitosa de un Sistema de Clasificación Acciones Humanas en tiempo real , logrando el objetivo central de reconocer acciones a partir de información anatómica obtenida de una persona en movimiento. La solución final se fundamenta en un *pipeline* robusto que integra de manera coherente la Visión por Computador y el Aprendizaje Automático.

La consecución de un Accuracy, Precision, Recall y F1-Score perfectos de 1.0 para ambos modelos sobre el conjunto de prueba valida la alta capacidad discriminativa de las características diseñadas. Esta robustez es el resultado directo de aplicar la teoría de la cinemática corporal en la Fase 2, utilizando geometría vectorial para extraer 48 características biomecánicas (ángulos, velocidades y estadísticas temporales) que aíslan el patrón esencial del movimiento, independientemente de la posición o la talla del sujeto. Para la clasificación, se emplearon los modelos de *ensemble learning* Random Forest y XGBoost , que manejan eficazmente las relaciones no lineales y las interacciones de características inherentes a los datos cinemáticos.

Finalmente, el modelo **Random Forest** fue seleccionado para la implementación en tiempo real , no solo por su ligeramente menor número de errores (15 vs 20) , sino también por su eficiencia computacional en la inferencia y su robustez ante el ruido, características fundamentales para la aplicación operativa.

Propuestas de mejora

Los errores residuales en la clasificación se concentraron principalmente en la confusión entre acciones con transiciones similares, como "caminar-atras" y "girar" (9 instancias en ambos modelos) , o "sentarse" y "caminar-adelante" (7 instancias en Random Forest). Para mitigar estos errores de transición, se recomienda ajustar la ingeniería de características. Específicamente, se sugiere incluir una *feature* que mida el desplazamiento absoluto del centro de la cadera a lo largo de la ventana temporal. El desplazamiento es nulo en acciones estáticas como "sentarse" y significativo al "caminar", lo que podría resolver la confusión entre estas clases. Adicionalmente, la inclusión de la aceleración angular (la derivada de las velocidades ya calculadas) proporcionaría una señal más clara

sobre el *inicio* y el *final* de las acciones, mejorando la discriminación de las transiciones suaves que actualmente confunden a los modelos.

Impactos del Sistema

El prototipo funcional de anotación de video tiene implicaciones significativas en varias dimensiones, que deben ser consideradas para su escalabilidad.

Dimensión Social:

El sistema es crucial para el monitoreo de pacientes y la ergonomía industrial al evaluar posturas riesgosas, lo que constituye un beneficio directo en la salud y seguridad de las personas. No obstante, el uso de la tecnología de estimación de pose implica la captura y procesamiento de datos biométricos. Para mitigar los riesgos de privacidad y la percepción de vigilancia, es fundamental implementar la anonimización de los datos, procesando exclusivamente los *landmarks* y no el video crudo, y asegurar el consentimiento explícito del usuario.

Dimensión Económica:

El sistema tiene un alto valor comercial en la industria del deporte (análisis de rendimiento), el entretenimiento, y la robótica. El beneficio económico se deriva de la automatización de tareas de monitoreo que, de otra forma, requerirían supervisión humana constante. La alta *accuracy* de aproximadamente 1.0 se traduce en un alto retorno de inversión debido a la baja tasa de falsos positivos o negativos. El uso de librerías de código abierto y modelos eficientes como Random Forest mantiene bajos los costos de implementación.

Dimensión Ambiental:

La inferencia en tiempo real requiere un procesamiento constante (alrededor de 25-30 *FPS*). Para la implementación a gran escala, es vital optimizar el modelo para correr en *hardware* de bajo consumo (*edge devices*), en lugar de depender de servidores potentes. Esta optimización energética es crucial para minimizar la huella de carbono operacional del sistema y asegurar su sostenibilidad ambiental a largo plazo.

Dimensión Global:

La normalización espacial por distancia entre hombros hace que el sistema sea robusto a variaciones de estatura y posición, garantizando su aplicabilidad global en diferentes poblaciones. Sin embargo, en regiones con estrictas regulaciones de privacidad (como la GDPR en Europa), la recolección y el procesamiento de información biomecánica (datos sensibles) requieren protocolos de seguridad de datos reforzados y cumplimiento legal exhaustivo. El sistema debe ser adaptable a diferentes marcos regulatorios internacionales para garantizar su viabilidad comercial en diversos mercados.

Enlace al Video: IA (https://icesiedu-my.sharepoint.com/:f:/g/personal/1110367830_u_icesi_edu_co/IgCaMT9-kqBqQ7jnmh19L4y6AUi6KtB0rPN38CETVgiuHqg?e=AgKxpc)