

## Tarea 3

### Instrucciones

- La tarea se realiza en grupos de 3 o 4 personas. Los archivos relacionados a la tarea se encuentran en el TEC - Digital, en la sección de Documentos->Tareas->Tarea 3.
- Los archivos computacionales deben ser implementados en GNU Octave. Por cada archivo que no este documentado correctamente, se restaran 5 puntos de la nota final.
- Si alguna función o archivo computacional está incompleto o genera error al momento de compilar, entonces pierde el 75% del puntaje de la pregunta asignada.

### Parte 1: Restauración de Imágenes

- La primera parte de la tarea consiste en la técnica de restauración de imágenes (*inpainting*, en inglés).
- El algoritmo a desarrollar en la tarea se encuentra en el artículo *Fast Digital Image Inpainting*, desarrollado por los investigadores M. Oliviera, B. Bowen, R. MacKena y Y. Chang de la Universidad Estatal de New York. Este algoritmo se basa en la convolución de matrices.
- La parte escrita debe estar en un documento pdf con nombre **Tarea 3 - Parte 1**. La infografía y los resultados numéricos e imágenes deben aparecer en el documento, explicados con detalle y claridad.

### Preguntas - Valor Total: 45 puntos

- **Pregunta 1 [Valor 10 puntos]:** Leer el documento '`restauracion_de_imagen.pdf`' y desarrollar una infografía del tema de restauración de imágenes. **Sugerencias:** (1) Leer los documentos '`inforgrafia1.pdf`' e '`inforgrafia2.pdf`' para poder crear una infografía. (2) Para desarrollar una infografía en internet, pueden acceder a la página <https://www.visme.co/>.
- **Pregunta 2 [Valor 5 puntos]:** Escriba un pseudocódigo del algoritmo presentado en el artículo científico *Fast Digital Image Inpainting*, que se encuentra en el archivo '`fast_digital_image_inpainting.pdf`'. El pseudocódigo debe indicar los valores iniciales y valores resultantes del algoritmo.
- **Pregunta 3 [Valor 20 puntos]:** Implemente computacionalmente en GNU Octave el algoritmo presentado en el artículo científico *Fast Digital Image Inpainting*, considerando la siguiente información:
  - (a) Los archivos a utilizar se encuentran en la carpeta con nombre **Parte 1 - P2**.
  - (b) La imagen original se encuentra en el archivo '`paisaje.jpg`' [Ver Figura 1(a)].
  - (c) La región a restaurar se encuentra en la imagen binaria '`marca.jpg`', donde las partes en color blanco son las regiones a restaurar [Ver Figura 1(b)].
  - (d) Ambas imágenes tienen el mismo tamaño ( $320 \times 662$ ).
  - (e) El código desarrollado en el archivo '`generar_imagen_restaurar.m`' genera la imagen que combina la imagen '`paisaje.jpg`' y '`marca.jpg`' [Ver Figura 1(c)].
  - (f) El objetivo de esta pregunta es realizar la restauración de la imagen de la Figura 1(c), utilizando el algoritmo del artículo científico *Fast Digital Image Inpainting* [Ver Figura 1(d)].



Figura 1: Imágenes de la Pregunta 3 - Parte 1

- **Pregunta 4 [Valor 10 puntos]:** El algoritmo propuesto en el artículo *Fast Digital Image Inpainting* también permite eliminar ciertos objetos de una imagen [Ver Figura 2].

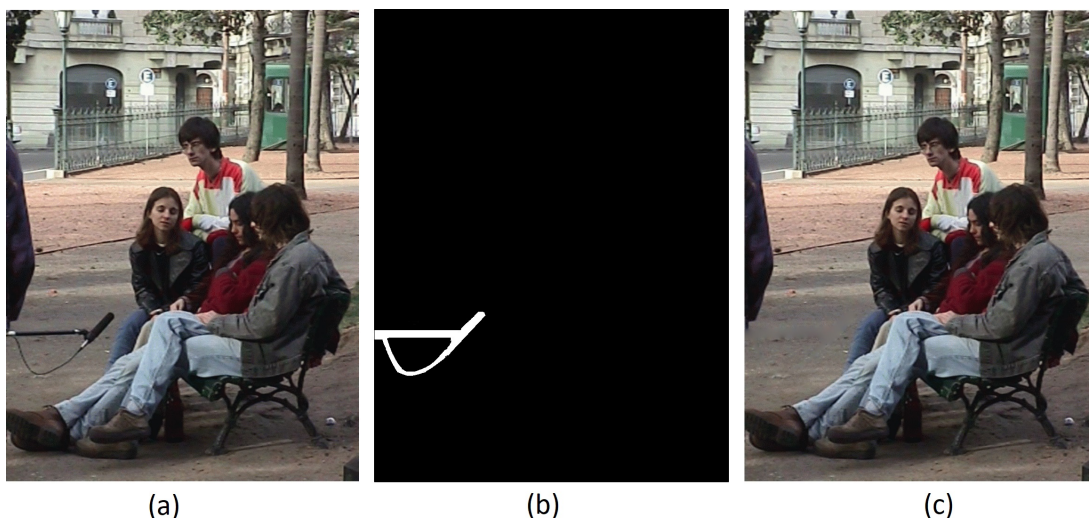


Figura 2: (a) Imagen Original. (b) Imagen Binaria del Objeto a Eliminar. (c) Imagen Reconstruida sin el Objeto.

**Cada grupo debe escoger una imagen a color donde aparezca, al menos, uno de los integrantes del grupo.** Además, en la imagen debe aparecer un objeto seleccionado arbitrariamente (un lápiz, un anillo, una cuerda, entre otros). El objetivo de esta pregunta es eliminar ese objeto de la imagen, utilizando el algoritmo propuesto en el artículo *Fast Digital Image Inpainting*. **Sugerencia:** El objeto debe identificarse claramente en la imagen, y no debe ser un objeto que abarque mucho espacio de la imagen.

## Parte 2: Croma o Clave de Color

- La segunda parte de la tarea consiste en una técnica llamada croma o clave de color.
- La parte escrita debe estar en un documento pdf con nombre **Tarea 3 - Parte 2**. Todos los resultados numéricos e imágenes deben aparecer en el documento, y explicado con detalle y claridad.
- La croma es una técnica audiovisual utilizada ampliamente tanto en cine, televisión y fotografía, que consiste en extraer un color de una imagen o vídeo (usualmente el verde) y reemplazar el área que ocupaba ese color por otra imagen o vídeo.
- Sea  $A$  una imagen a color de tamaño  $m \times n$  usando el modelo RGB. La imagen  $A$  tiene un fondo de color verde [Ver Figura 3(a)]. Sean  $A_r \in \mathbb{R}^{m \times n}$ ,  $A_g \in \mathbb{R}^{m \times n}$  y  $A_b \in \mathbb{R}^{m \times n}$  los canales rojo, verde y azul de la imagen  $A$ , respectivamente [Ver Figuras 3(b), 3(c) y 3(d)]. Como se observa en las imágenes de la Figura 3, un pixel  $(i, j)$  de  $A$  es de color verde si y solo si el pixel  $(i, j)$  de  $A_r$  y  $A_b$  son de tonalidad negra y el pixel  $(i, j)$  de  $A_g$  es de tonalidad blanca.

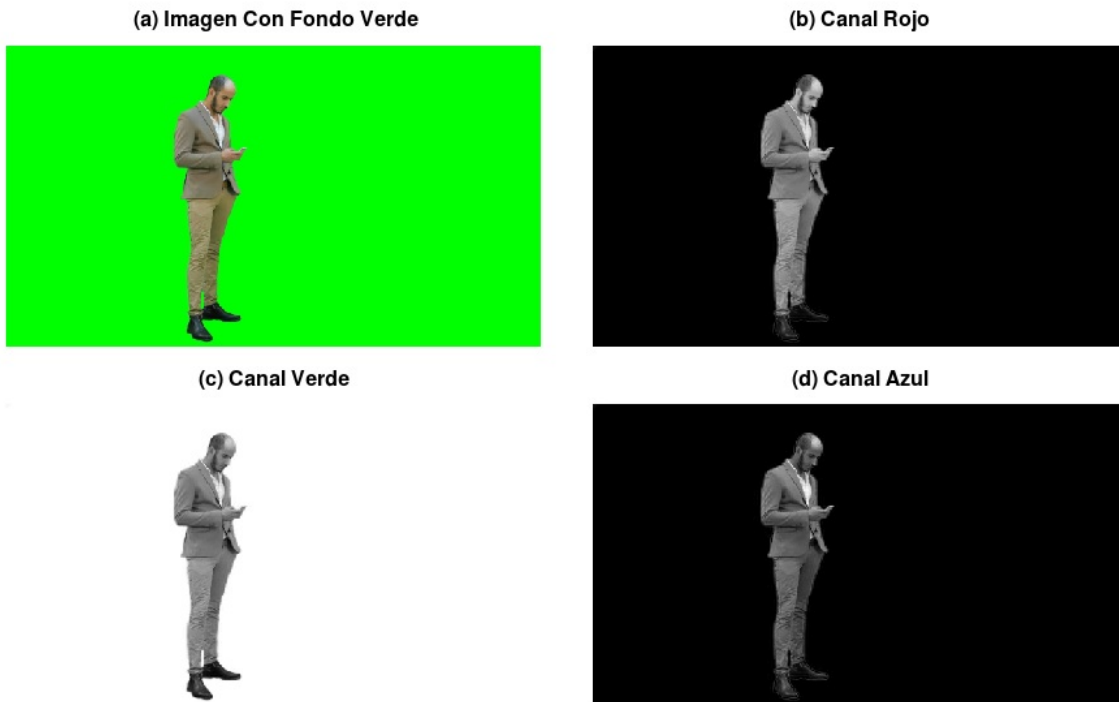


Figura 3: (a) Imagen con Fondo Verde. (b) Canal Rojo. (c) Canal Verde. (d) Canal Azul.

- Matemáticamente, lo anterior se puede representar de la siguiente manera:

$$A(i, j) = \text{verde} \iff A_r(i, j) = 0 \wedge A_g(i, j) = 255 \wedge A_b(i, j) = 0,$$

para todo  $i = 1, 2, \dots, m$  y  $j = 1, 2, \dots, n$ .

- Sea  $B$  una imagen a color de tamaño  $m \times n$ . La imagen  $B$  representa el nuevo fondo que se le quiere poner a la imagen  $A$ , en lugar del fondo en color verde [Ver Figuras 4(a) y 4(b)].
- Sea  $C$  la imagen de tamaño  $m \times n$  resultante de realizar el cambio de fondo en las partes verdes de la imagen  $A$ , utilizando la imagen  $B$  [Ver Figura 4(c)].

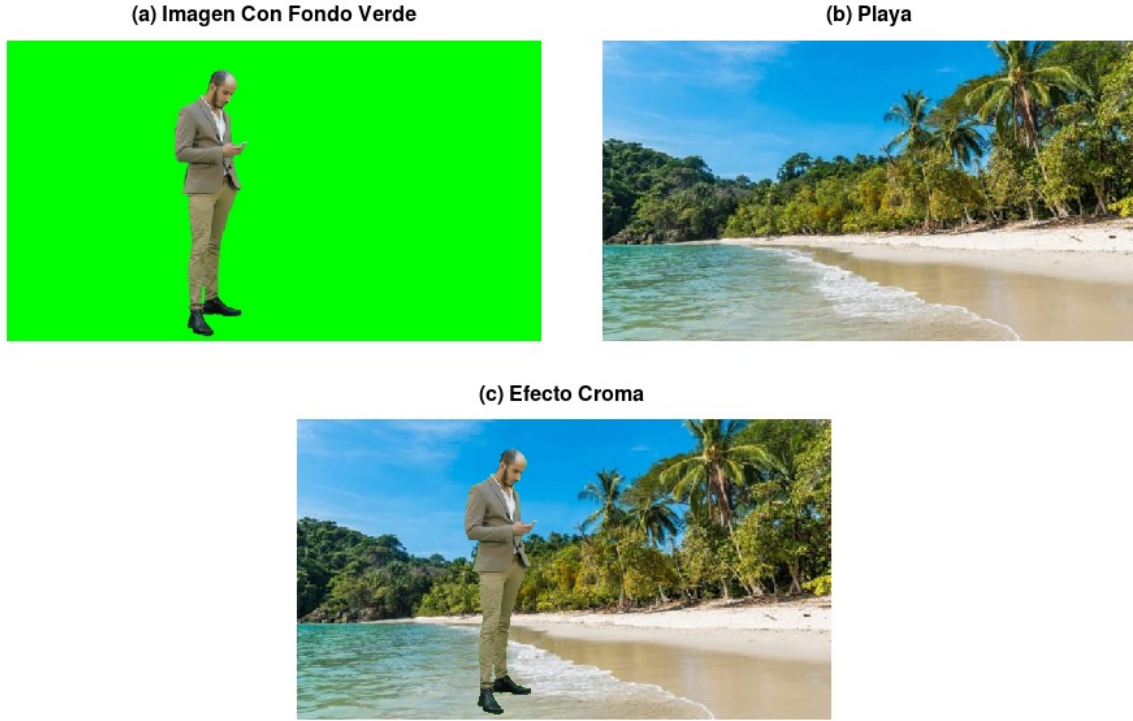


Figura 4: (a) Imagen con Fondo Verde. (b) Imagen del Nuevo Fondo. (c) Proceso de Croma.

- Un pseudocódigo para obtener la nueva  $C$  es el siguiente:

```

for  $(i, j) \in \{1, 2, \dots, m\} \times \{1, 2, \dots, n\}$  do
    if  $A(i, j) = \blacksquare$  then
         $C(i, j) = B(i, j)$ 
    else
         $C(i, j) = A(i, j)$ 
    end
end

```

## Preguntas - Valor: 55 puntos

- **Pregunta 1 [Valor 8 puntos]:** En GNU Octave, si se desea obtener las entradas que cumplen una condición booleana o acceder a algunas de las entradas de una matriz para cambiar su valor, no se recomienda utilizar un doble bucle **for**, ya que el tiempo de ejecución se incrementa significativamente. Para evitar este problema en esta parte de la tarea, se recomienda usar la estructura de programación que tiene GNU Octave. Esta pregunta tiene la finalidad de explicar como utilizar esta estructura. Para eso, considere la matriz de 3 dimensiones  $A \in \mathbb{R}^{4 \times 4 \times 3}$ , donde

$$A(:, :, 1) = \begin{pmatrix} 5 & 10 & 15 & 20 \\ 4 & 8 & 12 & 16 \\ 3 & 6 & 9 & 12 \\ 2 & 4 & 6 & 8 \end{pmatrix}, \quad A(:, :, 2) = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 1 & 3 & 5 & 7 \\ 2 & 4 & 6 & 8 \\ 1 & 4 & 7 & 10 \end{pmatrix}, \quad A(:, :, 3) = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 5 & 5 & 5 & 5 \\ 10 & 10 & 10 & 10 \\ 15 & 15 & 15 & 15 \end{pmatrix}.$$

- (a) **[Valor 4 puntos]**: Leer y ejecutar el archivo 'obtener\_indices\_matriz.m'. Utilizando la matriz  $A$ , implemente un *script* en GNU Octave que permita obtener una matriz de 2 dimensiones  $B \in \mathbb{R}^{4 \times 4}$ , tal que  $B(i, j) = 1$  si  $A(i, j, 1) > 5$ ,  $A(i, j, 3) \geq 10$  y  $A(i, j, 2) < 5$ ; y  $B(i, j) = 0$  en caso contrario. El nombre del archivo debe ser 'parte2\_p1a.m'. **Observación:** No debe utilizar ningún bucle para llegar al resultado final.
- (b) **[Valor 4 puntos]**: Leer y ejecutar el archivo 'evaluar\_matrices.m'. Utilizando la matriz  $A$ , implemente un *script* en GNU Octave que permita obtener una matriz de 3 dimensiones  $C \in \mathbb{R}^{4 \times 4 \times 3}$ , tal que  $C(i, j, k) = -30$  si  $A(i, j, k) \leq 10$  y  $C(i, j, k) = 30$  si  $A(i, j, k) > 10$ . El nombre del archivo debe ser 'parte2\_p1b.m'. **Observación:** No debe utilizar ningún bucle para llegar al resultado final.
- **Pregunta 2 [Valor 20 puntos]**: Implemente computacionalmente en GNU Octave el proceso de croma para sustituir el fondo verde de la imagen 'fondo\_verde.jpg' por la imagen 'playa.jpg' [Ver Figuras 4(a) y 4(b)]. El nombre del archivo debe ser 'parte2\_p2.m'. **Observaciones:** (1) No deben utilizar ningún bucle para llegar al resultado final. (2) Utilicen un valor de tolerancia, ya que los valores negros y blancos en los canales RGB de la imagen 'fondo\_verde.jpg' no siempre son 0 y 255, respectivamente.
  - **Pregunta 3 [Valor 12 puntos]**: El resultado final de la imagen obtenida en la Pregunta 2 puede generar un color verde (no puro) en los bordes de la figura incrustada [Ver Figura 5(a)]. Esto se debe a la mezcla de colores en el borde entre el fondo verde y el objeto. Para reducir el color color verde del borde, se utiliza una imagen binaria que representa el borde de la imagen incrustada [Ver Figura 5(c)]. Luego, se aplica un proceso de restauración, utilizando como referencia la parte blanca de la imagen binaria [Ver Figura 5(b)].

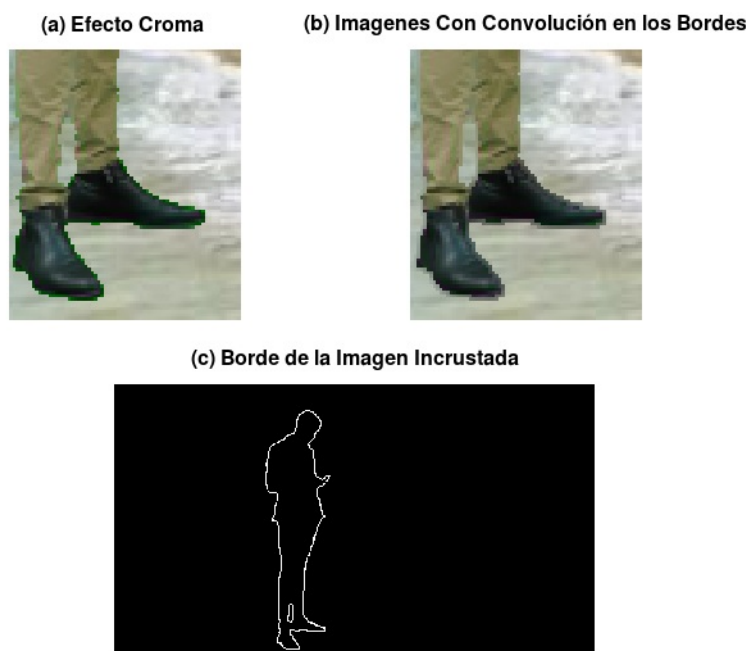


Figura 5: (a) Bordes de Color Verde después del Proceso de Croma. (b) Bordes de Color Verde Atenuados con Convolución. (c) Borde la Imagen Insertada.

En esta pregunta, deben modificar el código desarrollado en la Pregunta 2, de tal manera que se disminuya el color verde de los bordes de la imagen resultante. Para eso, utilice el algoritmo desarrollado en el Parte 1 de esta tarea. El nombre del archivo debe ser 'parte2\_p3.m'. **Observaciones:** (1) No debe utilizar ningún bucle para recorrer las entradas de las matrices. Solo se permite un bucle para eliminar el color verde del borde. (2) Para obtener el borde de la imagen, como se muestra en la Figura 5(c), se recomienda obtener una imagen binaria del canal verde de la imagen 'fondo\_verde.jpg', y luego utilizar algún operador morfológico para obtener solo el borde.



- **Pregunta 4 [Valor 15 puntos]:** Realizar el proceso de croma en un video, utilizando los siguientes archivos:

- 'video\_avion.mp4', el cual contiene un video de un avión con fondo verde [Ver Figura 6(a)].
- 'video\_cielo.mp4', el cual contiene un video del cielo [Ver Figura 6(b)].

El resultado debe ser un video con nombre 'video\_croma.mp4', que contiene al avión desplazándose en el cielo [Ver Figura 6(b)]. El nombre del archivo debe ser 'parte2.p4.m'. **Observación:** Deben utilizar el código desarrollado en la Pregunta 3.

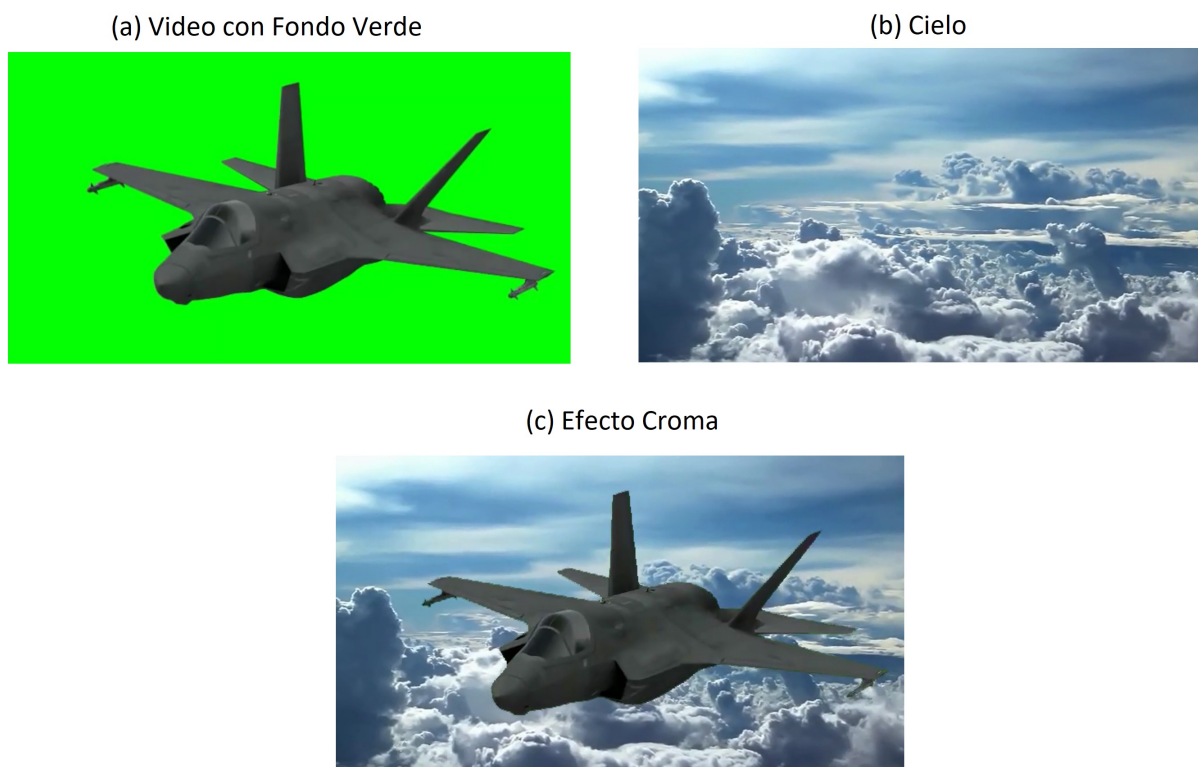


Figura 6: (a) Imagen del Video con Fondo Verde. (b) Imagen del Video con Nuevo Fondo. (c) Imagen del Proceso de Croma.

## Información de la Entrega

- **Fecha y hora límite:** Miércoles 5 de Agosto a las 11:59 pm.
- Los documentos deben estar guardados usando la siguiente estructura: Una carpeta principal con nombre **Tarea 3**. Dentro de esta carpeta, deben estar dos carpetas con nombres **Parte 1** y **Parte 2**. En cada una de estas carpetas estarán todos los archivos necesarios para el desarrollo de las preguntas mencionadas anteriormente.
- Deben enviar la carpeta **Tarea 3** en formato **zip** al correo [jusoto@tec.ac.cr](mailto:jusoto@tec.ac.cr), con el encabezado **Entrega Tarea 3 - PAID**. En el cuerpo del correo deben indicar el nombre completo de los miembros del grupo.
- **OBSERVACIÓN IMPORTANTE:** La entrega tardía de la tarea se penalizará con una reducción del 20% de la nota final, **POR CADA HORA DE ATRASO**.

## Defensa

- Cada grupo debe defender esta tarea frente al profesor. Para eso deben seleccionar un horario de la siguiente dirección electrónica:

<https://doodle.com/poll/59zy7hy4u4gkn6mw>

- Deben escribir el nombre (sin apellidos) de todos los miembros del grupo y seleccionar uno de los horarios disponibles.
- Todos los miembros del grupo deben estar presentes para defender cada una de las preguntas. Si un estudiante no está presente, entonces el estudiante perderá 35 puntos de la nota final.