

<b>EVALUACIÓN</b>	OBLIGATORIO 2	<b>GRUPO</b>	TODOS	<b>FECHA</b>	MAYO 2021
<b>MATERIA</b>	DISEÑO DE APLICACIONES				
<b>CARRERA</b>	ANALISTA EN TECNOLOGÍAS DE INFORMACIÓN / ANALISTA PROGRAMADOR				
<b>CONDICIONES</b>	<p>- Puntos: Máximo: 40 Mínimo: 1</p> <p>- Fecha máxima de entrega: 17/06/2021</p> <p>LA ENTREGA SE REALIZA EN FORMA ONLINE EN ARCHIVO NO MAYOR A 40 MB EN FORMATO ZIP, RAR O PDF.</p> <p><b>IMPORTANTE:</b></p> <ul style="list-style-type: none"> <li>- Inscribirse</li> <li>- Formar grupos de hasta tres personas.</li> <li>- Subir el trabajo a Gestión antes de la hora indicada, ver hoja al final del documento: "RECORDATORIO"</li> </ul>				

**El objetivo del presente trabajo obligatorio es modelar e implementar la primera versión prototipo de un juego de BINGO.**




---

**En esta primera versión el desarrollo se focalizará en:**

- 1) Precarga de información.
- 2) Aplicación para jugadores
- 3) Aplicación para administradores
- 4) Interfaz gráfica de escritorio

**Descripción general del dominio del problema:**

La mecánica del juego comprende un conjunto de al menos dos jugadores participando con uno o más cartones cada uno. Se sortea un número y luego cada jugador “anota” o “marca” si el número sorteado está presente en alguno de sus cartones. No puede existir un mismo número que se encuentre en más de un cartón (ni en el mismo). Ni pudiendo sortearse un número que no esté presente en alguno de los cartones de los jugadores que están participando en el juego. Pueden existir N juegos de BINGO simultáneamente.

## 1) Precarga de información.

**NO** es necesario implementar una interfaz de usuario para el mantenimiento de la información (salvo que se especifique explícitamente en algún caso de uso). El sistema deberá tener pre-cargada la información, de modo que al iniciarse ya cuente con un conjunto de datos definido.

La información que se debe precargar es:

- **Administradores:**
  - Información básica: cédula de identidad, contraseña, nombre completo, email.
- **Jugadores:**
  - Información básica: cédula de identidad, contraseña, nombre completo y saldo.
- **Parámetros de configuración** (incluye los valores con los que debe entregarse el programa):
  - Cantidad de filas del cartón = 3
  - Cantidad de columnas del cartón = 3
  - Cantidad máxima de cartones por jugador = 3
  - Cantidad de jugadores por juego = 3.
  - Valor del cartón = 100
  - Figuras habilitadas = Línea, Diagonal, Perímetro, Cartón Lleno.

Tener en cuenta:

- Estos valores pueden/deben variarse en el momento de testing/corrección.
- En todas las pruebas sólo se considerarán los siguientes límites de información:
- Cantidad de filas y columnas del cartón: entre 2 y 10
- Cantidad máxima de cartones por jugador: entre 1 e ilimitado
- Cantidad de jugadores por juego: entre 2 e ilimitado
- Valor del cartón: entre 1 e ilimitado
- Figuras: al menos una habilitada, máximo las 4 definidas hasta el momento.

**Cartón:** Un cartón es un conjunto aleatorio de números enteros positivos sin repetidos. Cada número se ubica en una posición única y fija dentro del cartón.

La cantidad de números en juego se calcula como la cantidad de números por cartón (cantidad de filas \* cantidad de columnas) multiplicada por la cantidad de cartones en juego.

La cantidad de cartones en juego es igual a la sumatoria de los cartones de cada jugador.

El cartón debe estar lleno. Ejemplos:

10	55
66	1

2 filas y 2 columnas

5	77	13	7
---	----	----	---

1 fila y 4 columnas

22	44	67
32	56	2
6	21	17
3	4	5

4 filas y 3 columnas

Los valores de cantidad de filas y cantidad de columnas determinan la forma del cartón.

**En un juego todos los cartones tienen la misma forma.**

**Cartones por jugador:** En un juego de BINGO cada jugador podrá participar con uno o más cartones. Este valor determina la cantidad máxima de cartones con que pueden participar los jugadores en un juego.

**Cantidad de jugadores:** Este valor determina la cantidad de jugadores que participan en cada juego.

**Valor del cartón:** Indica el valor que debe debitarse de su saldo a cada jugador, por cada cartón con que participe en un juego.

**Figuras:** Indica cuáles figuras están habilitadas para el juego y cuáles no. Los jugadores deberán anotar números de manera de formar alguna de las figuras habilitadas.

Las figuras son:

- *Línea:* Todos los números de una fila o columna en alguno de sus cartones.
- *Diagonal:* Todos los números de una diagonal en alguno de sus cartones. (Solo en caso de que el número de filas del cartón se igual al número de columnas)
- *Perímetro:* Los números anotados forman un rectángulo (o cuadrado) en el perímetro de alguno de sus cartones. Ej:

55	98	77	89
33	66	20	99
1	2	32	18
76	8	22	50

- *Cartón lleno:* Logra anotar todos los números de alguno de sus cartones.

## 2) Aplicación para Jugadores

### *CU: Ingresar a la aplicación*

Usuario	Sistema
1. Ingresar cédula, contraseña y cantidad de cartones con los que desea participar.	2. Ejecuta CU: Jugar al Bingo
	3. Fin caso de uso.

Cursos alternativos:

\*No existe en el sistema un Jugador cuyo nombre de usuario y contraseña coincidan con los ingresados. Acción: Mensaje: “Acceso denegado” y se vuelve a iniciar el caso de uso.

\* Un jugador solo puede participar de un bingo a la vez. En tal caso mostrar mensaje: “El jugador <cédula> ya está participando del bingo”

\*La cantidad de cartones no es un número mayor a cero. Acción: Mensaje: “Indique con cuantos cartones desea jugar” y se vuelve a iniciar el caso de uso.

\*La cantidad de cartones es un número mayor a la cantidad máxima de cartones con los que se puede participar. Acción: Mensaje: “No puede participar con más de X cartones” (siendo X la cantidad máxima de cartones) y se vuelve a iniciar el caso de uso.

\*El saldo del jugador no es suficiente para comprar la cantidad de cartones especificada. Acción: Mensaje: “Saldo insuficiente” y se vuelve a iniciar el caso de uso.

**IMPORTANTE:** Un jugador tiene saldo suficiente cuando su saldo es mayor o igual al triple del valor de un cartón multiplicado por la cantidad de cartones con los que participa.

El diagrama muestra una interfaz de usuario con el título "Ingreso jugador". Debajo del título, hay tres campos de entrada etiquetados como "Cedula:", "Contraseña:" y "Cant. Cartones:". Cada campo tiene un recuadro rectangular adyacente para ingresar el dato. Debajo de estos campos, hay un botón rectangular con el texto "Ingresar".

### *CU: Jugar al Bingo*

La mecánica del juego comprende un conjunto de al menos dos jugadores participando con uno o más cartones cada uno. Se sortea un número y para cada jugador el sistema “anota” o “marca” el número en el cartón correspondiente. Luego del sorteo cada jugador deberá indicar si desea continuar participando del juego.

El juego termina cuando hay un ganador. (\*5)

Usuario	Sistema
	1. Muestra el nombre de usuario del jugador y el número de juego (*1)
	2. Muestra mensaje “Esperando inicio del juego”.
	3. Espera para iniciar el juego a que entren tantos jugadores como lo indica la “cantidad de jugadores por juego”.
	4. Sortea un número del bolillero. (*2)
	5. Muestra el monto del pozo del bingo (*3) Muestra el saldo actual del jugador. Muestra los nombres de usuario <b>de los otros jugadores</b> que están participando del juego. Muestra los nombres de las figuras habilitadas. Muestra el último número sorteado. Muestra la secuencia histórica de números sorteados. Muestra los cartones del jugador. (*4) Si el jugador anotó el número sorteado muestra la palabra ¡Anotó!
	6. Si hay un jugador ganador ir al punto 12
7. Indica si desea continuar jugando o si abandona el juego.	8. Si el usuario indica que desea abandonar el juego ir a punto 14.
	9. Muestra la palabra “Esperando ...”
	10. Espera que todos los jugadores indiquen si continúan jugando o no.
	11. Si no hay un jugador ganador continúa el caso de uso en el punto 4.
	12. Muestra el nombre del jugador ganador. (*5) Muestra el nombre de la figura con la que ganó el jugador ganador. (*6) Muestra el monto del pozo ganado por el jugador ganador. (*7) Acredita el monto ganado al jugador ganador y debita el saldo de los demás jugadores participantes (*8)
	13. Muestra el saldo actual del jugador.
	14. Fin caso de uso

<< nombre del usuario >> << numero de juego >>

<< saldo del jugador >>

<< numero sorteado >>

<< monto del pozo >>

<< Estado del juego (Esperando inicio.../Anoto/Esperando...) >>

<< cartones >>

<< lista de jugadores >>

<< figuras habilitadas >>

Seguir jugando

Abandonar

<< secuencia de numeros sorteados >>

**(\*1) Mostrar nombre de usuario del jugador y el número de Juego.**

El nombre de usuario del jugador y el número de juego se muestra en el título de la ventana del jugador. El primer juego comienza con el número 1 (y se incrementa en 1 sucesivamente)

**(\*2) Bolillero.** Si por ejemplo hay 100 números en juego esto indica que “dentro del bolillero” están las bolillas del 1 al 100. **La cantidad de números en juego se calcula como la cantidad de números por cartón (cantidad de filas \* cantidad de columnas) multiplicada por la cantidad de cartones en juego. La cantidad de cartones en juego es igual a la sumatoria de los cartones de cada jugador.**

Durante un juego un mismo número no puede salir sorteado dos veces.

Tampoco puede sortearse un número que no esté presente en alguno de los cartones participantes en el juego. Tener presente que los jugadores podrán retirarse en medio de un juego.

### (\*3) Monto del Pozo:

El pozo del juego se calcula como la cantidad de cartones en juego multiplicada por el valor del cartón. Este valor no se modifica hasta que finaliza el juego. El valor final del pozo dependerá de la figura que anote el jugador ganador.

### (\*4) Mostrar cartones del jugador.

Cada cartón tendrá N números. Siendo N = cantidad de filas por cantidad de columnas.

Cada número se asigna al cartón eligiendo **un número aleatorio** entre 1 y la cantidad de números en el bolillero. No puede asignarse un mismo número en un cartón ni tampoco en dos cartones diferentes.

Los cartones del jugador se muestran en la interfaz de usuario (ventana) del jugador. El jugador debe poder visualizar fácilmente todos sus cartones. Los números del cartón deben estar alineados horizontal y verticalmente.

Los números anotados en el cartón deberán mostrarse de forma destacada, por ejemplo, resaltados en negrita o con un color diferente a los números sin anotar.

### (\*5) Un jugador gana el juego cuando:

a) Los demás jugadores se retiran.

b) Logra anotar números formando alguna de las figuras habilitadas para el juego.

(\*6) **Figuras:** Según la configuración realizada en los datos de prueba, en un juego pueden estar habilitadas desde al menos una a las cuatro figuras definidas hasta el momento.

### (\*7) Pozo ganado:

El pozo inicial del juego se calcula como la cantidad de cartones en juego multiplicada por el valor del cartón. O sea que al iniciarse el juego cada jugador paga al pozo por la cantidad de cartones que “compró” al ingreso del juego.

Si un jugador se retira del juego no deberá pagar ningún monto **extra** al pozo.

Si un jugador gana con la figura **Línea**, el resto de los jugadores que quedan en juego tampoco deberán pagar ningún monto **extra** al pozo.

En cambio, si un jugador gana con la figura **Diagonal** el resto de los jugadores que quedan en juego deberán pagar al pozo un monto **extra** equivalente al valor de medio cartón base (0.5) multiplicado por la cantidad de cartones con los que participó cada uno.

Si un jugador gana con la figura **Perímetro** el resto de los jugadores que quedan en juego deberán pagar al pozo un monto extra equivalente al valor de un (1) cartón multiplicado por la cantidad de cartones con los que participó cada uno.

Y si un jugador gana con la figura **Cartón Lleno** el resto de los jugadores que quedan en juego deberán pagar al pozo un monto extra equivalente al valor de dos (2) cartones multiplicado por la cantidad de cartones con los que participó cada uno.

En caso de formar dos figuras al mismo tiempo solo se tendrá en cuenta la que dé más rédito al jugador, o sea la que “pague más”.

### (\*8) Acreditar el pozo al ganador y debitar el saldo a los que no ganaron:

El monto que debe restarse al saldo de cada jugador que no ganó es el monto total que cada jugador pagó al pozo. (inicial + extra) (**ver \*7**).

**Importante:** El débito del saldo del jugador se realiza de forma efectiva al finalizar el juego, para asegurar que no se debitará de su saldo dinero si el juego se interrumpe por algún error inesperado.

No obstante, el saldo del jugador al iniciar el juego y durante el transcurso del mismo debe mostrarse actualizado (como si se hubiese restado de su saldo el dinero apostado) de forma que el usuario visualice claramente en todo momento el dinero que tiene.

### 3) Aplicación para administradores

Esta aplicación es utilizada sólo por los administradores. Los casos de uso disponibles en esta aplicación son los siguientes:

#### *CU: Ingresar a la aplicación*

Usuario	Sistema
1. Digita cédula de identidad y contraseña.	2.Ejecuta CU Monitorear juegos
	3. Fin caso de uso

Cursos alternativos:

2. No existe en el sistema un usuario cuyo nombre y contraseña coincidan con los ingresados.

Acción: Mensaje: “Acceso denegado” y se vuelve a iniciar el caso de uso.

Diagrama de la interfaz de ingreso de administrador. El título es "Ingreso administrador". Hay dos campos de entrada: "Cedula:" y "Contraseña:". Debajo de los campos hay un botón "Ingresar".

#### *CU: Monitorear juegos*

Usuario	Sistema
	2.Muestra una lista con los juegos de BINGO (*1)
3. Opcionalmente el usuario selecciona un juego e indica que desea ver sus detalles	4. Muestra los detalles del juego seleccionado (*2)
	3. Fin caso de uso

(\*1) La información a mostrar por cada juego es:

“Nº” + número de juego + “Estado:” + Esperando o Iniciado o Finalizado + “Jugadores:” + cantidad de jugadores



(\*2) La información a mostrar como detalles del juego es:

Número de juego

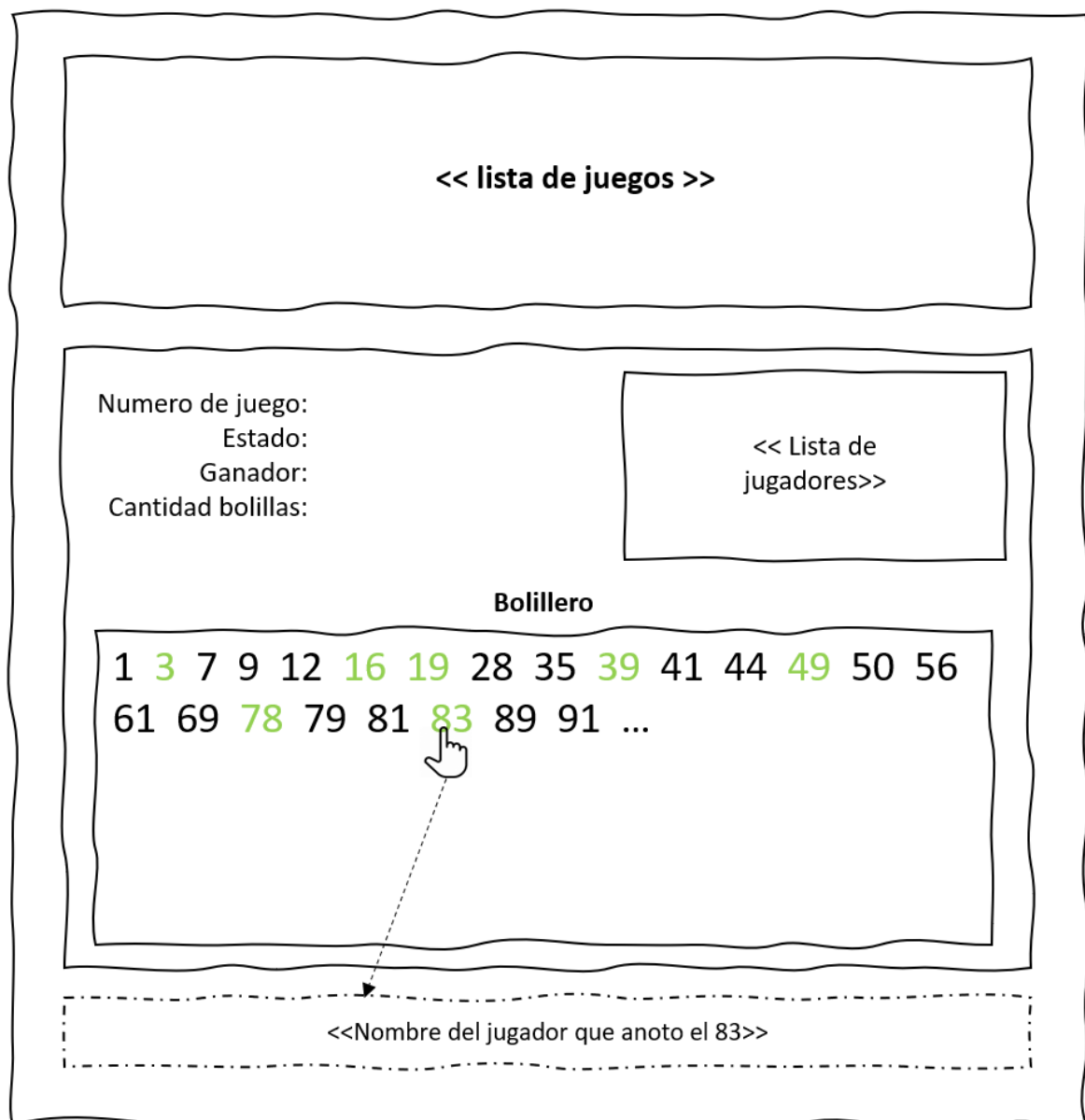
Estado

Ganador (si hay)

Cantidad de bolillas sorteadas

Nombres de jugadores participantes

Se debe mostrar gráficamente cada uno de los números del bolillero del juego, indicando si ya fueron sorteados o no. En caso de haber sido sorteados se deben marcar con un color diferente y al dejar puntero del mouse sobre el número o al hacer click sobre el número , debe mostrarse el nombre completo del jugador que anotó el número.



## 4) Interfaz gráfica

El objetivo de este prototipo es emular una situación en la cual cada usuario interactúa con el sistema desde una computadora diferente.

Para emular esta situación, se deberá implementar una ventana general para testing de la aplicación desde la cual se podrán lanzar instancias de las aplicaciones:

Jugadores: inicia con CU: Ingresar a la aplicación

Administradores: inicia con CU: Ingresar a la aplicación

**IMPORTANTE:** La información de *todas* las ventanas debe obligatoriamente actualizarse de manera automática, sin necesidad de que el usuario indique que desea actualizar la información.

### Requerimientos de diseño para esta versión:

1) Maximizar la modularidad y claridad del código. Para esto utilice el indicador que dice que ningún método debería tener más código que el que se puede visualizar en una pantalla.

2) Minimizar la duplicación de código. Evitar métodos o porciones de código que realizan la misma tarea.

3) División física de las clases en paquetes.

4) División lógica.

5) Uso del patrón de diseño “Fachada”.

6) Experto

7) Utilizar una arquitectura M.V.C.

8) Utilizar polimorfismo donde corresponda.

9) Utilizar manejo de excepciones.

---

### Notas

- Las posibles omisiones, ambigüedades o contradicciones que surjan del estudio de los requerimientos detallados en este documento serán analizadas y corregidas en clase durante el curso.

## Se pide entregar

**\*Implementación del sistema en Java con interfaz de usuario gráfica cumpliendo con todos los requerimientos funcionales y de diseño solicitados.**

### **\*2 Diagramas de Clases:**

- Un diagrama de clases conceptual modelando el dominio del problema.
- Un diagrama de clases de diseño (o más de uno si lo considera apropiado) que incluya a todas las entidades que participan en la solución. (Iu + Lógica)

**NO es necesario especificar atributos y métodos en los diagramas.**

### **\*Autoevaluación:**

Descripción breve de aquellos requerimientos funcionales o de diseño que faltan o no funcionan correctamente y/o pueden ser mejorados y el motivo. Teniendo en cuenta las reglas de calificación (ver más abajo) auto califique su trabajo. La calificación debe incluir un detalle por áreas y una calificación general.

Se adjuntan los puntos para cada requerimiento:

## Distribución del Puntaje

Concepto	Pts.
Funcionalidad: Implementación de la funcionalidad solicitada en Java	<b>(-39)</b>
Diagrama conceptual de domino.	<b>2</b>
Diagrama de diseño	<b>2</b>
Claridad y reúso del código	<b>2</b>
División lógica	<b>6</b>
Experto	<b>6</b>
Fachada y arquitectura	<b>3</b>
MVC	<b>6</b>
Observador y manejo de eventos	<b>4</b>
Manejo de excepciones	<b>3</b>
Strategy	<b>5</b>
Autoevaluación	<b>1</b>
<b>Total:</b>	<b>40</b>

## Defensa

La defensa del trabajo intenta:

\*Evaluar el conocimiento general de los integrantes del grupo sobre la solución propuesta. Todos los integrantes deben conocer toda la solución.

\*Evaluar el aporte individual al trabajo por parte de cada uno de los integrantes del equipo.


\*El mecanismo de defensa se determinará al momento de la entrega, pudiendo ser el mismo escrito o en el laboratorio. En ambos casos se verificará el aporte al trabajo de ambos integrantes.

# RECORDATORIO: IMPORTANTE PARA LA ENTREGA

- **Obligatorios**

La entrega de los obligatorios será en formato digital online, a excepción de algunas materias que se entregarán en Bedelía y en ese caso recibirá información específica en el dictado de la misma.

Los principales aspectos a destacar sobre la **entrega online de obligatorios** son:

1. Ingresá al sistema de Gestión.
2. En el menú, selecciona el ítem “Evaluaciones” y la instancia de evaluación correspondiente, que figura bajo el título “Inscripto”.
3. Para iniciar la entrega hacé clic en el ícono: 
4. Ingresá el número de estudiante de cada uno de los integrantes y hacé clic en “Agregar”. El sistema confirmará que los integrantes estén inscriptos al obligatorio y, de ser así, mostrará el nombre y la fotografía de cada uno de ellos. Una vez agregados todos los integrantes, hacé clic en “Crear equipo”.

**Cualquier integrante podrá:**

- **Modificar la integración del equipo.**
- **Subir el archivo de la entrega.**

5. Seleccioná el archivo que deseas entregar. Verificá el nombre del archivo que aparecerá en la pantalla y hacé clic en “Subir” para iniciar la entrega. Cada equipo (hasta 3 estudiantes) debe entregar **un único archivo en formato zip o rar** (los documentos de texto deben ser pdf, y deben ir dentro del zip o rar). El archivo a subir debe tener **un tamaño máximo de 40 mb**

Cuando el archivo quede subido, se mostrará el nombre generado por el sistema (1), el tamaño y la fecha en que fue subido.

6. El sistema enviará un e-mail a todos los integrantes del equipo informando los detalles del archivo entregado y confirmando que la entrega fue realizada correctamente.
7. Podés cerrar la pestaña de entrega y continuar utilizando Gestión o salir del sistema.
8. La **hora tope para subir el archivo será las 21:00** del día fijado para la entrega.
9. La entrega se podrá realizar desde cualquier lugar (ej. hogar del estudiante, laboratorios de la Universidad, etc).
10. Aquellos de ustedes que presenten alguna dificultad con su inscripción o tengan inconvenientes técnicos, por favor contactarse con el Coordinador o Coordinación adjunta **antes de las 20:00 hs.** del día de la entrega, a través de los mails [gervaz@ort.edu.uy](mailto:gervaz@ort.edu.uy), [alamon@ort.edu.uy](mailto:alamon@ort.edu.uy) y [terra@ort.edu.uy](mailto:terra@ort.edu.uy), o telefónicamente al 29021505 - int 1156 (de 8:00 a 14:00 hs) y 1436 (de 17:30 a 20:00 hs).

Si tuvieras una situación particular de fuerza mayor, debes contactarte con suficiente antelación al plazo de entrega, al Coordinador de Cursos ([gervaz@ort.edu.uy](mailto:gervaz@ort.edu.uy)) o al Secretario Docente ([paulos@ort.edu.uy](mailto:paulos@ort.edu.uy)).