

GUÍA PARA PROGRAMAR EXPERIMENTOS EN PSICOLOGÍA

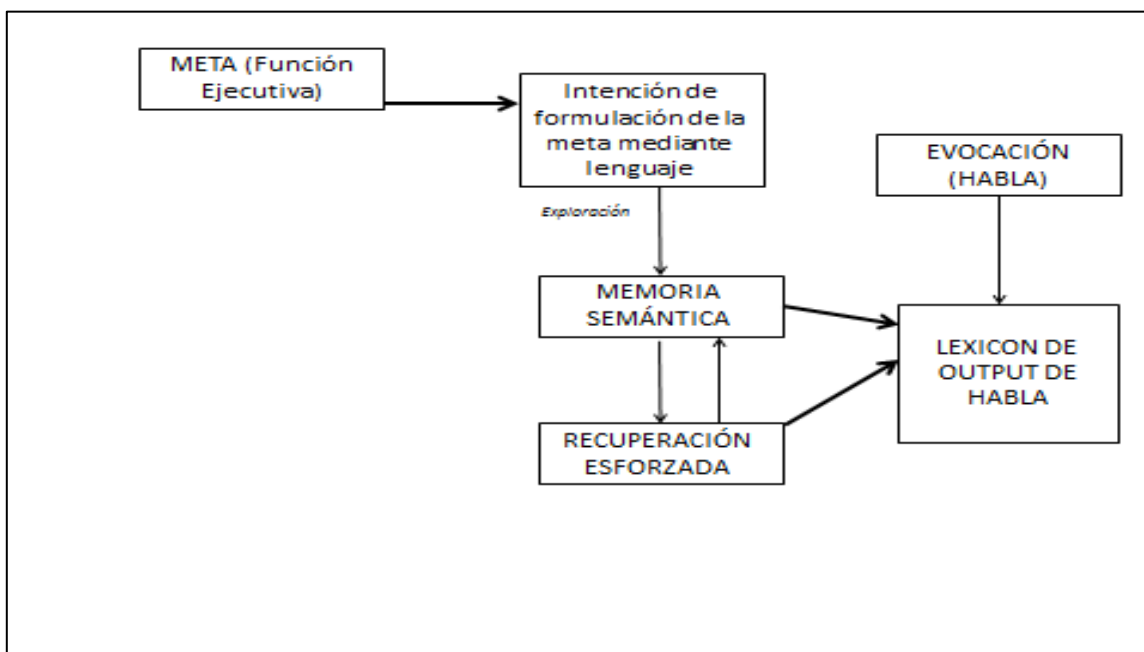
Julián Marino

Fernando Luna

PARTE 1: ASPECTOS CONCEPTUALES DE LA PROGRAMACIÓN DE EXPERIMENTOS EN PSICOLOGÍA

1. ¿Por qué programar experimentos en Psicología?

Quizá no se encuentre entre las principales expectativas de un estudiante de Psicología la idea de aprender técnicas y diseños de programación de experimentos en un ordenador. Sin embargo, es una herramienta sumamente útil para obtener datos relacionados con los modelos teóricos que aprende. Esto se aplica con rigurosidad a la Neuropsicología Cognitiva, que requiere de evidencias obtenidas por esta vía para validar una proposición conceptual. La programación de un experimento crea una situación conductual, de allí que los datos que obtiene se denominan 'datos conductuales'. En Neuropsicología Cognitiva también se piden datos de neuroimágenes y datos fisiológicos, que se relacionan con los datos conductuales. En conjunto aportan a los diagramas de flujo mediante los que tradicionalmente se representan sus modelos teóricos. A continuación, un modelo neurocognitivo de evocación conceptual. Se define 'evocación conceptual' como la creación 'esforzada' de una persona de una definición mediante el lenguaje hablado (Badre & Wagner, 2007). Se aprecia que involucra diferentes capacidades y funciones ejecutivas:



Un diagrama de flujos cognitivo debe ser factible de ser transducido a una situación conductual. El experimentador cognitivista crea la situación conductual mientras mantiene un diagrama de flujos que le guía y orienta en sus procedimientos. Hay que matizar como se interpreta actualmente un diagrama de flujos cognitivo. En sus orígenes representaba la manera en qué efectivamente la información 'transcurría' por módulos cognitivos, implicaba una secuencia temporal (Vega Rodríguez, 1984).

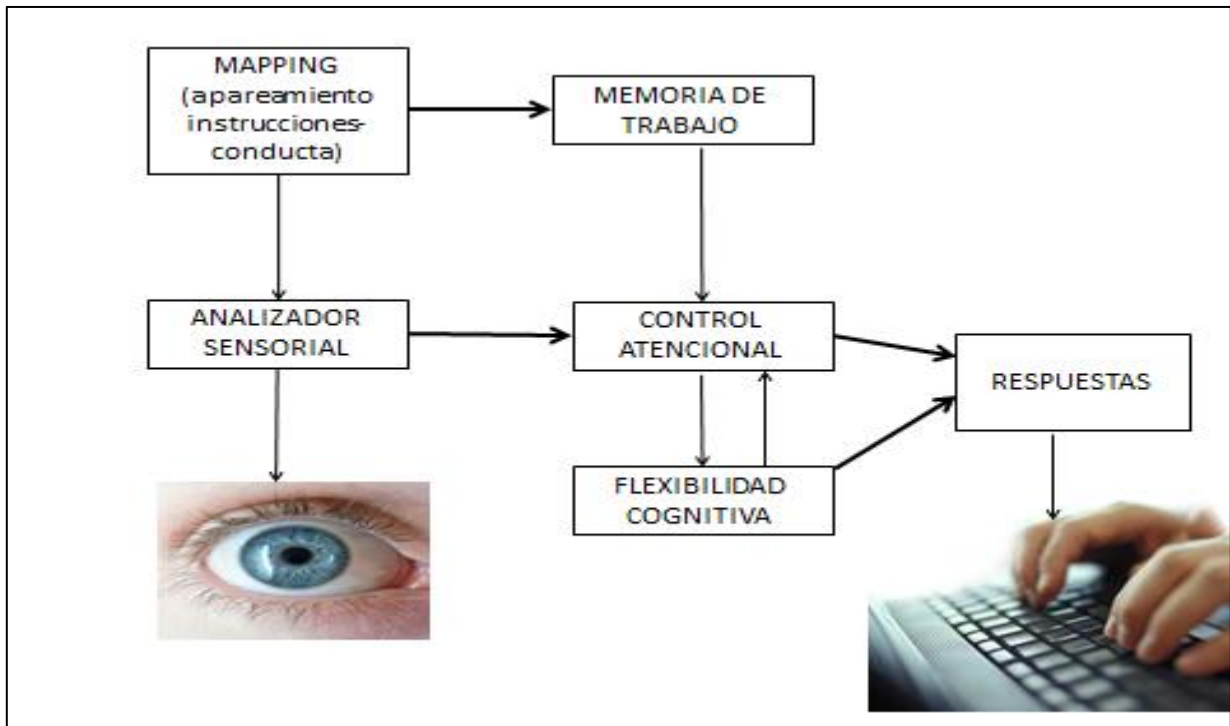
Actualmente se asemejaría a un modelo que grafica posibles relaciones de influencias entre variables. Como se discute en el apartado 8 de la parte 1, la aparición de los modelos de redes neuronales y de actividad en paralelo cuestionaron con severidad el uso de diagramas de interpretación secuencial (Rumelhart & McClelland, 1987). Sin embargo, no hay un desprendimiento total de esa concepción, los centros nerviosos y sus conectividades están relacionados con módulos de tratamientos específicos de información, que se vinculan a capacidades cognitivas, por ejemplo memoria de trabajo fonológica o almacén semántico. Se podría sugerir que los diagramas de flujo se interpretan a medias como relaciones entre variables, de manera similar a los modelos que se crean mediante ecuaciones estructurales y a medias como flujos de información secuenciales.

Veamos un ejemplo: Se le pide a un psicólogo cognitivo que diseñe una situación conductual donde esté fuertemente comprometida la flexibilidad cognitiva y crea el siguiente experimento: Se presentan rostros de mujeres y hombres que pueden ser jóvenes o de edad avanzada. Los rostros tienen a su alrededor un marco verde y otras veces un marco morado. Se le pide a la persona que si un rostro tiene un marco verde responda de acuerdo al sexo, presionando la tecla 'x' si se trata del rostro de un hombre y la tecla 'n' si se trata del rostro de una mujer. En cambio, si el marco es morado se le pide que responda de acuerdo a la edad, presionando la tecla 'z' si es una persona joven, independientemente del sexo, y la tecla 'm' si el rostro pertenece a una persona de edad avanzada. Se considera la siguiente secuencia:



Si se presentan estos dos eventos la persona debería presionar consecutivamente las teclas 'n' y 'm', que serían las respuestas correctas. La secuencia del experimento completo será notablemente más extensa, los atributos de los estímulos admiten varias combinaciones y sus 'cruzamientos' admitirían la presentación de 120 rostros (eventos), 60 de marco verde y 60 de marco morado. En ambos tipos de enmarcaciones habría 15 mujeres jóvenes, 15 mujeres de edad avanzada, 15 varones jóvenes y 15 varones de edad avanzada. La secuencia de aparición de los eventos sería aleatoria, y a la persona se le pediría que trate de responder adecuadamente (ajustándose a las reglas) y de la forma más rápida posible.

¿Cuál es el diagrama de flujo teórico que acompaña este experimento?



Se aprecia que se estimula la actividad de las Funciones Ejecutivas, ya que están comprometidos la memoria de trabajo, el control atencional y la flexibilidad cognitiva (Miyake et al., 2000). El diagrama de flujo indica que se presenta un 'mapping' en modalidad visual (las instrucciones se brindan en pantalla), se trata de formar un 'mapeo nervioso' que conecte los estímulos que serán presentados con las respuestas que deben emitirse. Mientras lee la presentación del experimento la persona almacena en su memoria de trabajo las instrucciones y las aparea con las respuestas. La memoria de trabajo es un subsistema de memoria, operativa y a corto plazo, que consolida las asociaciones que guiarán la conducta durante el experimento. Esta parte es crucial ya que si el 'mapping' se realiza de forma descuidada afectará negativamente el rendimiento (la persona 'olvidará' que tecla correspondía a qué condición, y el objetivo era evaluar la flexibilidad cognitiva, y no la capacidad de recuerdo).

Una vez producido el mapping comienza la presentación de los eventos. Se define eventos como las realidades que controla el experimentador para provocar procesamiento de información. En este caso son rostros enmarcados en diferentes colores. Para procesar la secuencia de rostros y marcos y sus combinaciones se espera que la persona active recursos de la red fronto- parietal de control, que es la más compleja y ejecutiva de las redes atencionales (Posner & Petersen, 1989). En el diagrama de flujo se establece que se utilizará la información almacenada durante el mapping en la memoria de trabajo mientras se ejerce el control estratégico sobre los estímulos que son presentados. ¿Por qué se califica como 'control estratégico'? Porque los cambios de eventos se suceden a cierta velocidad y con un grado de complejidad que la persona buscará desarrollar estrategias para automatizar sus respuestas. Esto se relaciona con la actividad dorsolateral prefrontal, la actividad cingulada anterior y la creación de modificaciones en el procesamiento sensorial en las regiones parietales (Botvinick, Braver, Barch, Carter, & Cohen,

2001). En sentido conductual se mejora el procesamiento de los estímulos y la velocidad para detectar rasgos y luego se conecta este 'refinamiento perceptual' con la ejecución de la conducta motora de presión de la tecla adecuada (de forma progresiva se generan estrategias asociativas particulares que automatizan que el color verde implica sexo y el color morado edad).

El éxito en la tarea estará determinado no solo por el sostén del control perceptual de los estímulos, sino también por la capacidad de flexibilidad que tenga la persona para pasar de un tipo de estímulo a otro. La flexibilidad cognitiva se define como la capacidad de alternancia cognitiva entre diferentes secuencias de información (Crone, Richard Ridderinkhof, Worm, Somsen, & Van Der Molen, 2004). Tal como indica el diagrama de flujo en este experimento el mapping ingresa de modo verbal a través de la lectura, luego se procesan estímulos que activan el control ejecutivo y el nivel de rendimiento en control atencional, flexibilidad cognitiva y memoria de trabajo determinará la calidad y velocidad de las respuestas motoras que se emitirán.

El modelo o diagrama de flujo indica que el nivel de control atencional y el nivel de flexibilidad cognitiva predicen directamente el rendimiento en el experimento, y que la memoria de trabajo realiza una predicción indirecta. Esto permite realizar inferencias de rendimiento en patologías neuropsicológicas. Se señaló que el control atencional está influido por la capacidad de memoria de trabajo que fue activada cuando se realizó el mapping. Si una persona padece una enfermedad degenerativa en la estructura hipocámpica tendrá escasos recursos de memoria de trabajo y es probable que su rendimiento sea defectuoso independientemente del esfuerzo en control atencional (Mitchell, Johnson, Raye, & D'Esposito, 2000). También existen patologías asociadas a la rigidez cognitiva (falta de flexibilidad) como la esquizofrenia. El mapping podría haber sido exitoso y la persona conservar intactos los recursos de memoria de trabajo y de control atencional, sin embargo la falta de flexibilidad característica de la esquizofrenia interactúa con el control atencional y se realizaría un procesamiento 'rígido' de los estímulos, resultando costoso para la persona pasar de un tipo de estímulo a otro. Por ejemplo se demorará significativamente si aparecen dos fotografías de mujeres jóvenes consecutivas acompañadas de marcos de diferente color, que implica un cambio de criterio de respuesta ante un estímulo similar en un amplio porcentaje. La flexibilidad cognitiva fue asociada a la actividad de las porciones dorsolaterales prefrontales de la corteza cerebral y su deterioro causaría un rendimiento final defectuoso en esta tarea.

En esta primer parte se destaca que se programan experimentos en Psicología para obtener datos conductuales, y se crea una situación conductual que está acompañada de un soporte conceptual representado por un diagrama de flujo de información. ¿Por donde transita la información que se representa en los diagramas de flujo cognitivos? Por las redes nerviosas de la corteza y subcorteza cerebral cuya funcionalidad está representada teóricamente por capacidades (lenguaje, memoria, atención, cognición espacial) funciones ejecutivas (procesos que coordinan las capacidades cognitivas y las dirigen hacia metas) y por modalidades sensoriales de entrada (visual, auditiva, gustativa, kinestésica, táctil, olfativa). Si el experimento no está relacionado con un modelo teórico no se trata de un experimento, sino de un juego o serie de ensayos sin sentido conceptual.

Los experimentos que se programan por ordenador tienen la posibilidad de ser coordinados con la obtención de datos neurofisiológicos y de neuroimagen. El software que presentamos se encarga de unificar en líneas de tiempo que transcurren simultáneas y en paralelo la información que arriba desde el experimento programado (eventos) y desde la tecnología que se conecta con el cuerpo humano y obtiene señales de su funcionamiento nervioso. Se aprecia la interface simultánea entre datos fisiológicos o de neuroimagen y datos conductuales. Esta es en sí misma una poderosa razón para aprender a programar en ordenador los experimentos, de otra manera la coordinación de datos es inalcanzable.



2. La recolección de datos conductuales

Las personas están constantemente inmersas en situaciones conductuales. La existencia de cada persona entre otras personas supone una vasta riqueza de intercambios simbólicos anidados en contextos que admiten diferentes niveles de análisis. Una situación conductual frecuente es una persona que espera en una larga fila para cobrar un cheque en un banco. El avance progresivo de la fila, la respuesta emocional de tedio, mantener los papeles de la transacción en la mano, los pensamientos distractivos implican un flujo de información en el sistema cognitivo, sin embargo los estímulos que recibe y las respuestas no están controladas. En general, la mayor parte de las situaciones conductuales se encuentran libres de observación y medición, los estímulos que provocan procesamiento de información en las personas no están siendo investigados como variables.

Por el contrario, el investigador que recoge datos conductuales estructura y controla la situación, organiza estímulos, el entorno y recoge respuestas luego de programar una serie de eventos. ¿Cuáles son los datos conductuales? Los tests neuropsicológicos tradicionales obtienen

rendimiento (puntajes finales) a través de conductas que implican destrezas en pruebas de lápiz y papel, manipulación de estímulos específicos como mazos de cartas o respuestas a ordenadores. Los tests se suelen utilizar en ámbitos clínicos, se tienen en cuenta tanto el rendimiento final como el proceso de ejecución (cómo lo hizo) y suelen ser interpretados a través de valores normativos (baremos) que indican la posición relativa que ocupó un puntaje en comparación con una población de referencia.

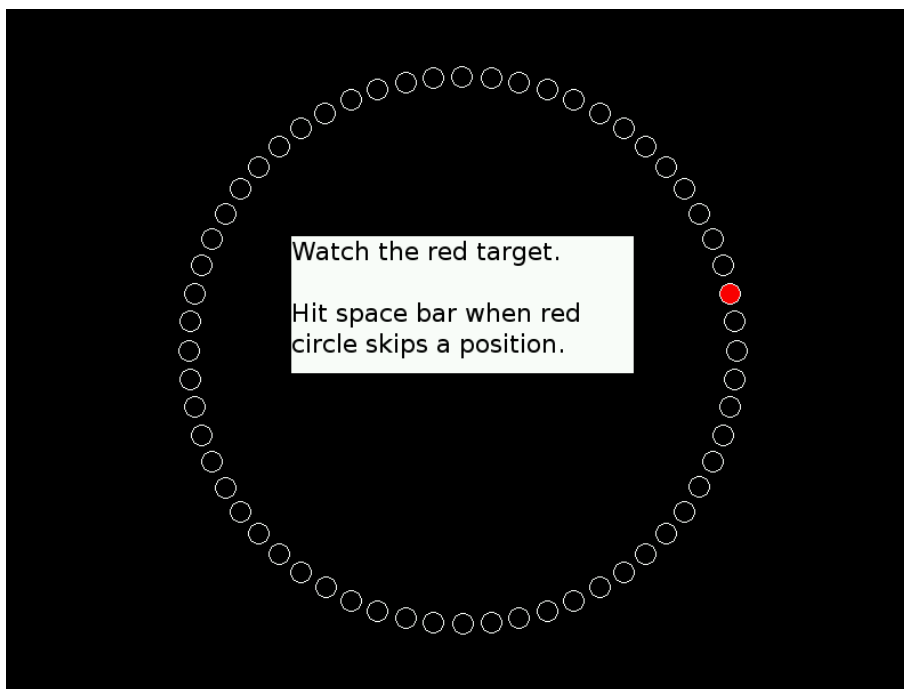
Una tarea consiste en paradigmas que se aplican en ordenadores y también obtiene datos conductuales, por ejemplo las tareas go/ no go, los paradigmas de flancos, los conflictos tipo Stroop o Simon. La presentación de estímulos sigue una lógica estructurada, por ejemplo en las tareas go no go debe haber un estímulo ante el que debe emitirse una respuesta y otro de características parecidas que demanda la abstención de respuestas, estas estructuras globales se comparten en las numerosas aplicaciones de go/ no go, pero las formas de presentación concretas suelen resultar muy variadas (el tipo de estímulo que determina si se deberá a presionar o no la tecla puede ser un número, una figura geométrica, un tipo de palabra).

En un experimento cognitivo programado en ordenador el diseño se incluye en una investigación que pone a prueba una hipótesis que cuestiona conocimientos que han sido establecidos o busca generar nuevas evidencias sobre poblaciones o efectos desconocidos. En numerosas ocasiones la innovación consiste en una variación de una tarea o paradigma, en otras ocasiones la creación del experimento sigue lineamientos más 'originales'. Cuando la tarea o el experimento han sido programados en ordenador se obtienen dos tipos de respuesta, la respuesta concreta que emitió (por ejemplo, qué tecla presionó) y cuánto tiempo demoró en hacerlo. La respuesta concreta se relaciona con el mapping presentado y en función de tal relación se categoriza como respuesta correcta o incorrecta, se suele utilizar el anglicismo '*accuracy*' para caracterizar esta dimensión de la respuesta. En algunos casos se programan tareas donde las respuestas no son correctas ni incorrectas sino que interesa conocer solo qué tecla presionó, la respuesta pertenecería a categorías diferentes y en su mayoría forman parte de adaptaciones de tareas autoinformadas que han sido pasadas a lenguaje de programación para facilitar su aplicación. No nos centraremos demasiado en estas aplicaciones.

En la programación de experimentos debe estar claro qué respuesta se considerará correcta, cuál será incorrecta y determinar en qué condiciones hay ausencia de respuestas. En la parte 2 se aprenderá como programar este aspecto que es crucial en los experimentos en ordenador. Otra dimensión de la respuesta conductual es el tiempo de reacción (TR) y se mide en milisegundos (1 segundo es igual a 1.000 milisegundos). La notable ventaja de la programación de experimentos mediante software especializado es la exactitud de la adquisición de TR, cuestión que resulta casi imposible en los tests tradicionales. Algunos tests de lápiz y papel como el Trail Making Test utilizan el tiempo como medida dependiente, pero la medición se realiza en segundos y resulta más 'gruesa' que los TR aquí descriptos (el evaluador suele tener un cronómetro que enciende y detiene con sus dedos).

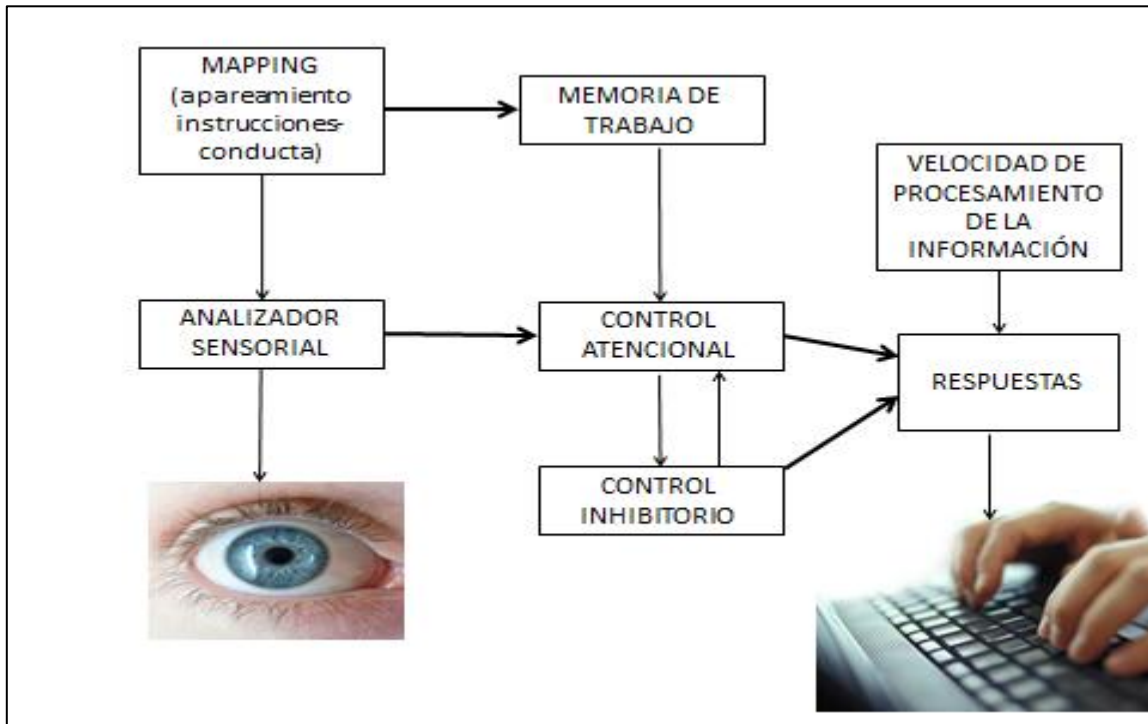
Los experimentos suelen presentar los eventos de forma repetitiva debido a la varianza intrasujetos en las respuestas. Regresemos al ejemplo del apartado 1 cuando se presentaban rostros y marcos de colores variables. Supongamos que nos interesa conocer si se responde más rápidamente a la edad o al sexo dentro de las condiciones de las fotografías. Si se presenta solo un evento para cada condición se obtendría un resultado poco representativo de la variabilidad de respuestas que ofrece una persona, que por supuesto tendrá TR diferentes a lo largo de varios ensayos y también cometerá algunos errores. Para obtener un número significativo de eventos se necesitan varias respuestas bajo la misma condición (siguiendo el ejemplo, presentación de rostros de mujeres de edad avanzada con marcos verdes) y un número 'mágico' aceptable implica obtener al menos 30 respuestas por condición (también se suelen aceptar 15 por condición).

El investigador debe hacer algo que resulta complejo pero crucial para que su experimento tenga sentido teórico, formular las hipótesis en términos de que la mayor o menor cantidad de respuestas correctas y el mayor o menor tiempo de reacción tengan consecuencias teóricas. Esto es fundamental y se vincula con el respaldo conceptual que tiene cada experimento programado. Por ejemplo, en un experimento con drogas psicotrópicas se aplicó una tarea de visual tracking (rastreo visual) a personas que habían ingerido una dosis de Clonazepam y a personas que no la habían consumido. La tarea de rastreo visual programada fue la siguiente:



Se diseñó el experimento para que la luz roja 'salte' de círculo en círculo en la misma dirección que las agujas del reloj. La persona debía controlar el pasaje de la luz roja de un círculo a otro, y cuando se omitía un círculo consecutivo se debía presionar la barra espaciadora. Se consideraron la cantidad de respuestas correctas divididas en dos tipos, respuestas correctas de omisión (cuando la luz roja pasaba a un círculo consecutivo se debía abstener de responder) y respuestas de presión correctas (cuando se salteaba un círculo adyacente debía presionar la barra). Se

consideró el porcentaje de respuestas correctas y también se consideraron los tiempos de reacción en las respuestas de presión correctas (en general, cuando se tiene en cuenta TR se consideran sólo las respuestas correctas) y se compararon los dos grupos. El investigador trabajó con el siguiente diagrama de flujo de respaldo:

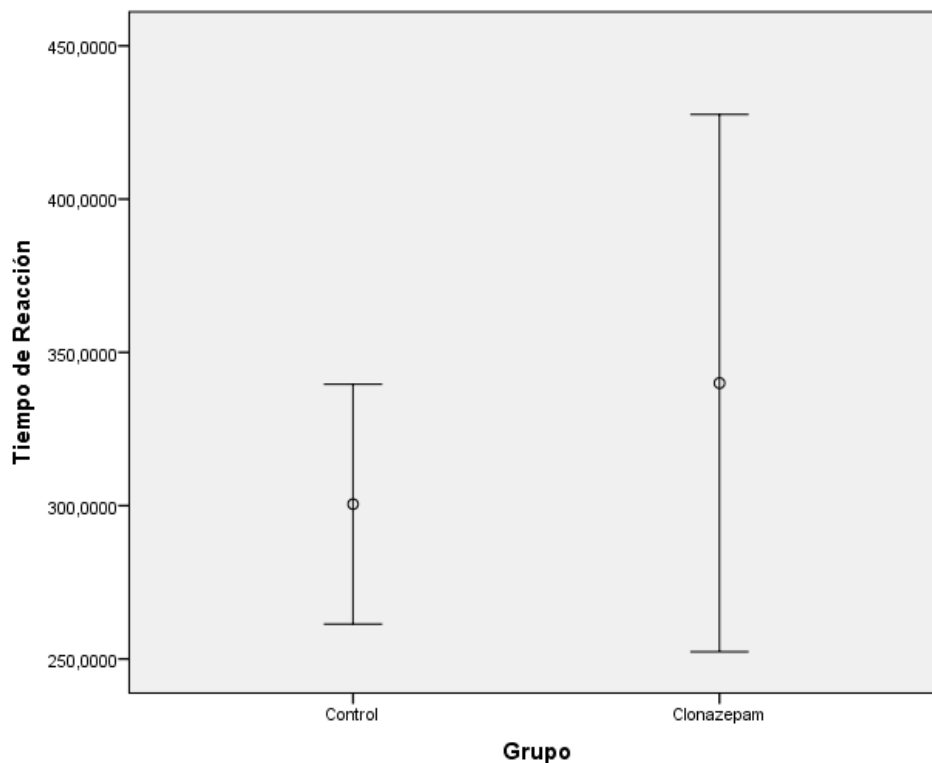


Se puede apreciar que es un experimento que activa el control inhibitorio, que es una función ejecutiva básica y que las respuestas (accuracy y TR) están influidas por la velocidad de procesamiento. En general, se considera que la velocidad es una capacidad general del sistema cognitivo, difícil de atribuir a una zona cortical en particular. Una pregunta que se podría formular es si en el experimento que presentaba rostros también habría que incluir la influencia de la velocidad de procesamiento de la información. Los modelos deben ser exhaustivos sobre los módulos y capacidades que intervienen en cada proceso y cada vez que hay una tarea que implique TR, la capacidad general de velocidad de procesamiento es muy probable que tenga peso predictivo. Que la medida conductual sea TR tiene una significación particular para las medidas psicológicas, y está relacionado con el fuerte rol que desempeñaría la capacidad de velocidad de procesamiento de la información. Determinar el porcentaje de explicación que la velocidad por sí sola tiene sobre cualquier medida de TR serviría para conocer cuánto de los resultados pueden atribuirse al proceso particular medido por el diseño.

Luego de revisar en la literatura las propiedades y efectos nerviosos del Clonazepam y observando el diagrama de flujo, el investigador conjeturó que las personas que habían consumido el psicotrópico tendrían menor velocidad de procesamiento de la información y que mantendrían un nivel normal de control inhibitorio y control atencional. También conjeturó que el mapping podría ser peor en las personas que consumieron Clonazepam, que tiene efectos paradójicos en la

memoria de trabajo. En personas con trastornos de ansiedad basales tal sustancia suele mejorar el rendimiento de memoria de trabajo porque aumenta la posibilidad de concentración, pero la tasa de prevalencia de esos trastornos era de 3.3% en la población y en sujetos escogidos al azar había poca probabilidad de incluir uno. Entonces se hipotetizó que quienes consumieron Clonazepam tendrían TR significativamente más prolongados que los individuos controles y también que cometerían mayor cantidad de errores, pero en esta medida la diferencia podría no resultar significativa, ya que la afección de Clonazepam sobre el control atencional sería entre leve y muy leve.

Finalmente se realizó el experimento en 30 personas, 15 ingirieron una dosis de Clonazepam de 1 miligramo una hora antes de la ejecución de la tarea y 15 personas recibieron una pastilla placebo que tenía la misma forma, color y presentación que Clonazepam pero químicamente no contenía ningún psicotrópico. Los resultados en tiempos de reacción fueron los siguientes:

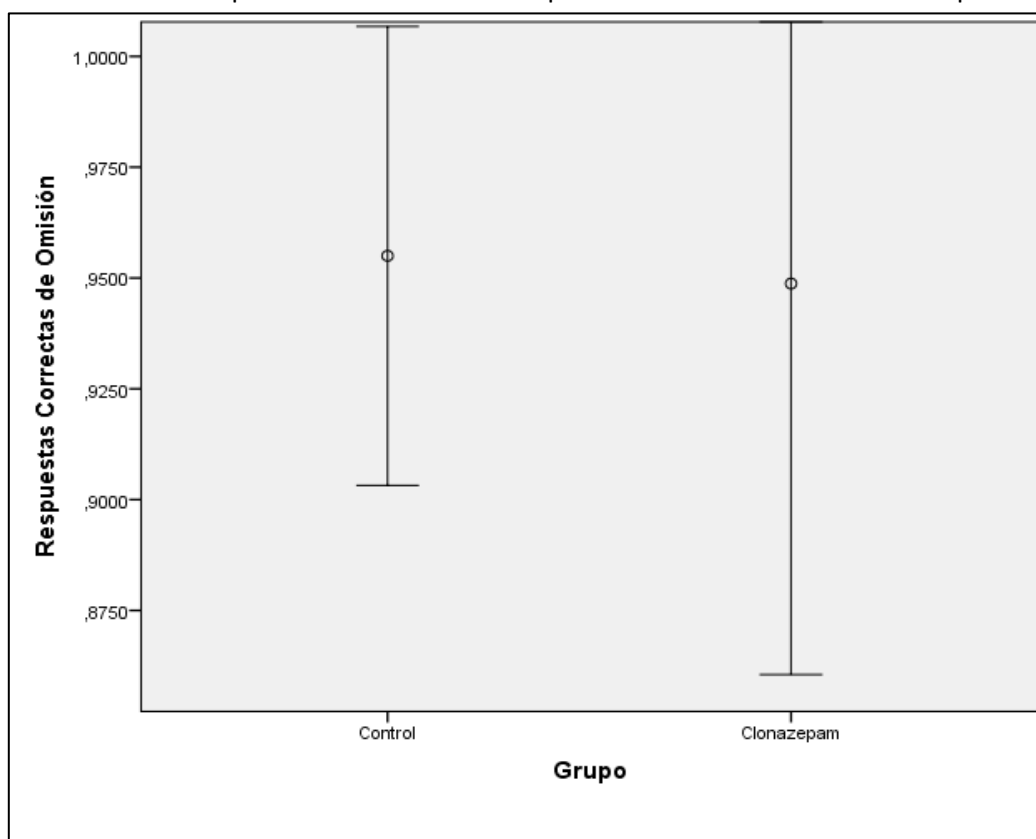


El gráfico superior informa sobre la media en milisegundos de rendimiento de cada grupo en TR y la desviación típica. La relación entre la media y la desviación entre los grupos es crucial para determinar si existió una diferencia significativa en el rendimiento entre los dos grupos. En el gráfico se advierte que los controles tuvieron una media cercana a los 300 milisegundos, fueron más rápidos que los que ingirieron Clonazepam (cerca de 350 milisegundos) pero, ¿fue esta diferencia suficiente para considerarse significativa? Eso se determina con pruebas de hipótesis inferenciales, como el análisis de varianza o las pruebas t de Student, entre otras. Los programas de ordenador como el que se verá en la parte 2 arrojan la distribución de los milisegundos que

demoró cada participante en cada ensayo, y el análisis estadístico se realiza con paquetes especializados, como SPSS o SAS.

Las barras indican también que la variabilidad en TR dentro de quienes consumieron Clonazepam fue considerablemente mayor que la variabilidad en los controles: ¿Implicaría esto una interacción con niveles de ansiedad basales que no fueron controlados? El investigador regresó a su diagrama de flujo y se percató que la influencia de la memoria de trabajo pudo haber sido central en la gran variabilidad de los consumidores de Clonazepam, ya que es una función que resulta modulada por trastornos psicopatológicos que podrían haber tenido las personas y que la sustancia habría modulado en su rendimiento. El investigador se percató que debía llamar nuevamente a los participantes y aplicarles una escala autoinformada de ansiedad como STAI para aumentar la validez de los resultados obtenidos.

A continuación se presentan los datos correspondientes a la exactitud de la respuesta de omisión:



Se aprecia que el porcentaje de respuestas correctas fue más alto para el grupo control que para el que ingirió la dosis aguda de Clonazepam, sin embargo, por inspección visual se advierte que esta diferencia estaría 'contenida' dentro de la variabilidad que describen las barras de desviación. Las respuestas de omisión significaban el rendimiento en control inhibitorio, las personas que ingirieron el fármaco presionaron la barra espaciadora con mayor frecuencia cuando no debían en comparación con las personas que sólo ingirieron el placebo, lo que implicaría cierta falla en esa capacidad ejecutiva. Sin embargo, si la diferencia resulta estadísticamente significativa también se

determina con una prueba de hipótesis inferencial. De forma intuitiva las apariencias indican que la diferencia sería estadísticamente despreciable.

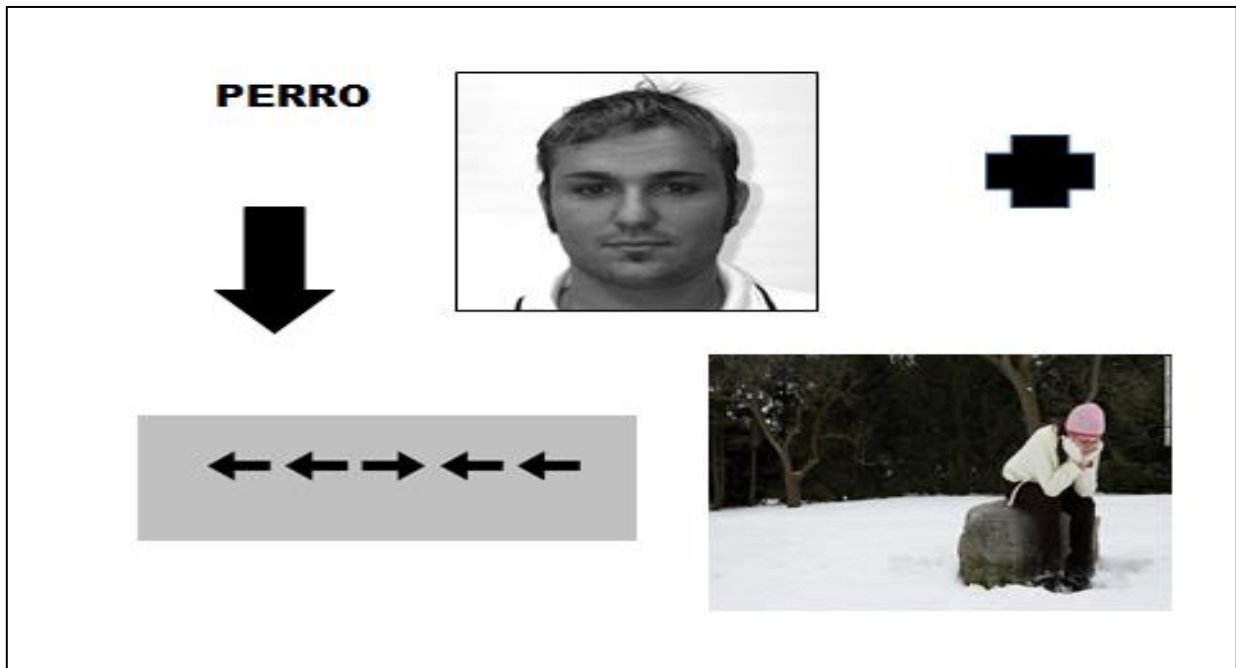
Programar un experimento permite obtener datos conductuales que comprenden 'accuracy' y tiempos de reacción. Sin la programación en ordenador esto resultaría imposible, más cuando se necesita una precisión de milisegundos. Las respuestas motoras ante los estímulos más habituales no suelen sobrepasar los 1.000 milisegundos, incluso las respuestas en las tareas atencionales clásicas, como la tarea ANT (Callejas, Lupiáñez & Tudela, 2004) suelen promediar los 600 milisegundos. En el ejemplo que se acaba de presentar, los TR son menores a medio segundo, y los datos deben recogerse con la máxima precisión. Se debe considerar que la diferencia de 50 milisegundos que fue hallada podría ser estadísticamente significativa, definir un problema conceptual, y se está en presencia de diferencias temporales que implican un veinteavo de segundo.

Cuando se presentan palabras es posible obtener respuestas más lentas ya que el procesamiento semántico suele ser más 'pesado' que el procesamiento perceptual de figuras como flechas o asteriscos. Como fue señalado anteriormente el 'accuracy' y TR deben ser utilizados para formular hipótesis teóricas, si se revisan las revistas científicas de psicología cognitiva se apreciará que la mayor cantidad de artículos con datos conductuales presentan sus afirmaciones conceptuales basados en diferencias en tales medidas. La cuestión clave y que revela maestría de dominio es establecer un planteo conceptual que culmine en un experimento donde por ejemplo tener TR más lentos implique que un módulo o función cognitiva está siendo interferida por otra, o que directamente no está participando en el proceso, o que el aumento en TR implique que hay una capacidad cognitiva que interviene en el proceso.

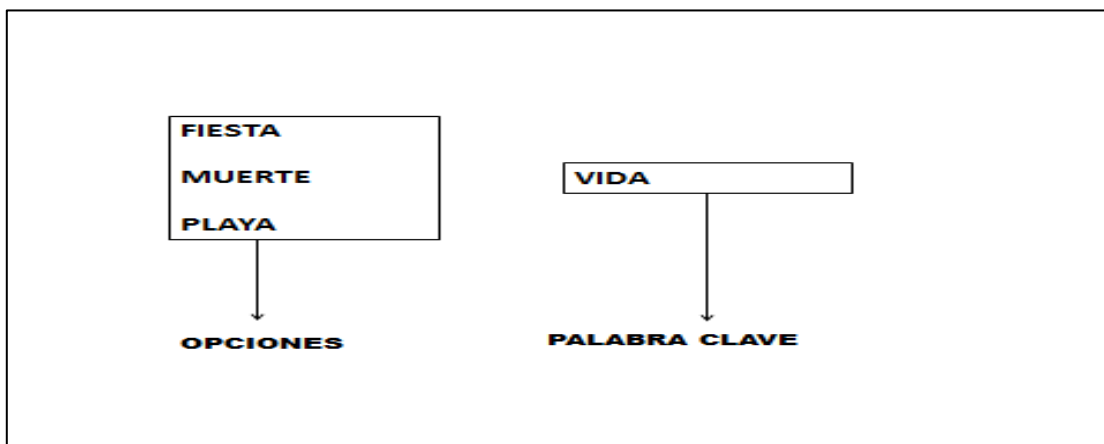
El uso de TR y 'accuracy' fue tradición de los experimentos cognitivos y su interpretación original se realizó dentro del marco de las teorías sustractivas de Donders y de factores aditivos de Sternberg. Los experimentos atencionales que se iniciaron en los años '50 del siglo veinte los emplearon y fundamentaron los modelos cognitivos de la atención y la memoria. Para una revisión crítica de estas metodologías clásicas de la psicología véase De Vega (1998) y el apartado 8 de la parte 1.

3. Los estímulos y el mapping

En la programación de experimentos en Psicología se presentan estímulos, están controlados, diseñados y planificados por el psicólogo experimental, que los aplica para activar el decurso de información que quiere someter a prueba. Los estímulos pueden ser imágenes (rostros, fotografías), palabras, sonidos, videos y combinaciones de cada uno de estos. A continuación se presentan ejemplos típicos de estímulos visuales presentados en experimentos psicológicos:



Las presentaciones estímulas implican un tiempo preestablecido de exposición y la determinación de las respuestas posibles. Sobre las presentaciones estímulas se crean y manipulan condiciones, que son dimensiones que organizan la lógica de presentación en dirección a cumplir con los objetivos experimentales. Por ejemplo, el investigador le presenta a una persona tres palabras opciones y una palabra clave, y establece que 5000 milisegundos es el tiempo máximo de respuesta. La persona debe decidir lo más rápido posible, dentro del margen de tiempo estipulado, cuál de las tres palabras opciones tiene un significado más cercano a la palabra clave. En el mapping el investigador estableció que si el participante decide que la palabra que está en la posición superior es la más relacionada por su significado con la palabra clave debe presionar la tecla 1, si cree que es la palabra del medio, presiona la tecla 2 y si cree que es la palabra inferior presiona la tecla 3.



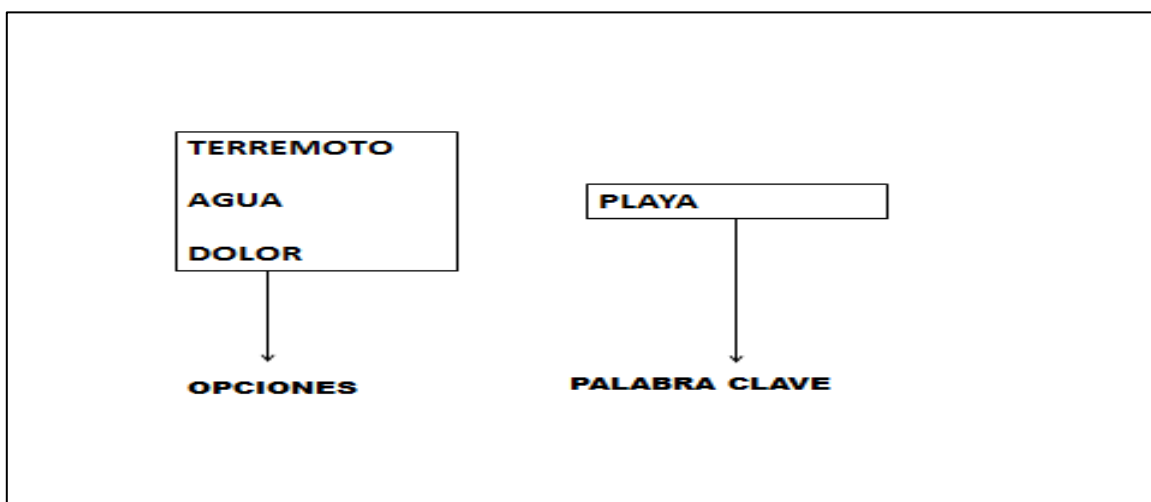
En el ejemplo de la figura superior la respuesta correcta sería la tecla 2, porque la palabra más relacionada con ‘vida’ por su significado es ‘muerte’ dentro de las tres opciones presentadas. ¿De qué manera este ejemplo sirve para ilustrar qué son las condiciones?

Se definió condiciones como dimensiones que organizan las presentaciones estímulas. En este caso, los estímulos son palabras, y en cada presentación estimular (le llamaremos ‘eventos’) aparecen tres palabras opciones (en la sucesión de eventos las palabras específicas van cambiando) y una palabra clave (va cambiando la palabra clave). En este evento hay dos opciones- palabra que tienen una valencia emocional positiva (FIESTA- PLAYA) y una opción- palabra que tiene valencia emocional negativa (MUERTE). La palabra clave tiene valencia emocional positiva (VIDA).

Si se considera que VIDA y MUERTE eran las más cercanas por su significado, se considera que hay una incongruencia afectiva entre la palabra clave y la respuesta correcta. En la jerga, se denomina habitualmente incongruencia cue (clave)- target (correcta- objetivo). Las palabras incorrectas (más lejanas a la palabra clave) se consideran distractores, y en este ejemplo de evento tienen similar valencia emocional que la palabra clave, por lo tanto el evento tiene congruencia cue- distractores. Quedaron así definidas las siguientes condiciones de la presentación estimular:

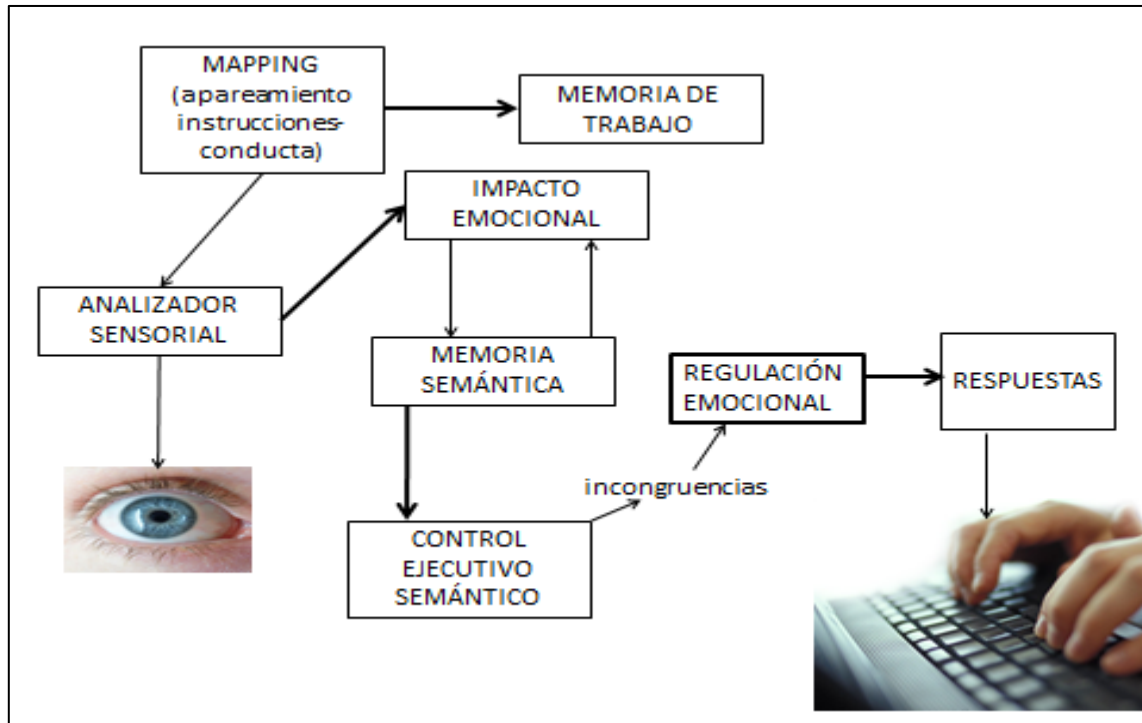
CUE (valencia emocional)	TARGET (valencia emocional)	DISTRACTORES (valencia emocional)
POSITIVO	POSITIVO	POSITIVO
NEGATIVO	NEGATIVO	NEGATIVO

¿Cuántas condiciones quedaron definidas? Ocho, ya que hay 2 x 2 x 2 posibilidades combinatorias. Por ejemplo, si se presenta un evento CUE POSITIVO- TARGET POSITIVO- DISTRACTORES NEGATIVO se podría presentar:



Se aprecia que la estructura estimular se repite (tres palabras opciones, una palabra cue, una de las opciones es el target). El evento pertenece a una nueva condición y para un experimento es

crucial que las condiciones tengan relevancia teórica. En este caso, la significación teórica de las condiciones se presenta en el siguiente diagrama de flujo:



Con la presentación de estímulos bajo condición de incongruencias emocionales (por ejemplo, incongruencias cue- target con congruencias cue- distractores) se aumenta el rol de la regulación emocional. Regulación emocional es un proceso que implica procesar un impacto emocional y controlarlo de forma ejecutiva, en este caso con utilización de información semántica (esta es una tarea diseñada para medir una forma de regulación emocional conocida como reappraisal o resignificación) (Ochsner & Gross, 2005). Esta presentación de diversas condiciones tiene sentido teórico: las condiciones con mayor congruencia impactan menos en la necesidad de regulación emocional que los ensayos incongruentes. En los capítulos posteriores y en la parte 2 se aprenderá a aplicar la lógica de condiciones mediante un elemento que adquirirá notable relevancia, los Atributos.

El mapping es fundamental porque organiza lo que debe hacerse ante la presentación de estímulos, realiza una conexión entre las secuencias que se presentarán y lo que debe ejecutar la persona como respuesta. Resulta clave la objetividad en la realización del mapping, se aprecia en el diagrama de flujo que está relacionado con la actividad de la memoria de trabajo durante el total de la tarea. Si una persona recibe mayor tiempo de mapping que otra, tendrá ventajas en la ejecución de sus respuestas, que no podrán ser atribuidas con seguridad a los efectos de las condiciones porque se sospecharía de un mejor procesamiento a causa de mejor mapping. Una técnica utilizada en programación es estandarizar el mapping mediante instrucciones escritas que aparecen en la pantalla del ordenador y mediante la realización de una ejercitación previa a los ensayos del experimento. Se suelen programar pantallas de texto con las instrucciones, la persona

las lee y presiona una tecla cuando considera que las ha comprendido, y luego realiza los ensayos previos. En estos casos, se suele utilizar una herramienta denominada feedback, que le informa a la persona si su respuesta fue la adecuada o no, incluso cuál fue su tiempo de reacción. Se suele advertir con un sonido chirriante o un pitido si la persona se equivoca, así se provoca alarma y se induce a la corrección de la performance.

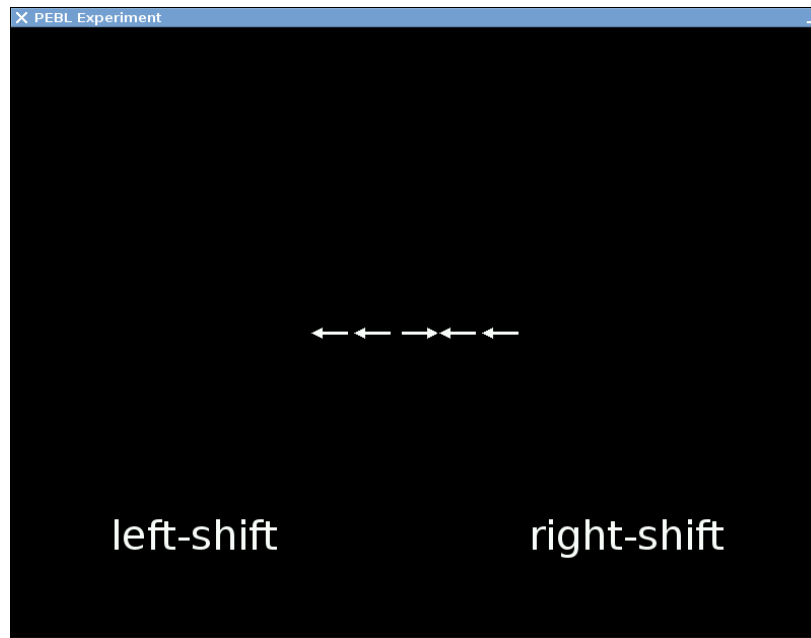
Una vez empezado el experimento, se suele advertir que no habrá más feedback. Tener un feedback (retroalimentación) en un ensayo previo ayuda a conocer el éxito del mapping que se ha realizado. Se aprende a responder de forma adecuada por moldeamiento, la persona realiza los apareamientos entre las secuencias estímulares y las respuestas. Se desaconseja incluir intervenciones verbales en el mapping por parte del investigador, ya que esto sería variable dadas las diferencias individuales (hay personas más tímidas, inhibidas, más verborrágicas), y entonces el rol de la memoria de trabajo no estaría controlado. En la parte 2 habrá una explicación concreta de cómo programar bloques en los experimentos (bloques de ensayos previos, bloque experimental 1, bloque experimental 2) y también como insertar la herramienta feedback y como quitarla.

4. La redacción del mapping

La claridad y la redundancia son virtudes de un buen mapping. Hay que recordar que las presentaciones estímulares se componen de secuencias de eventos repetitivos y el mapping indica que conducta debe ser ejecutada. Veamos el siguiente ejemplo, donde el mapping se presenta de esta manera:

A continuación se te presentará una serie de flechas (cinco). Debes concentrarte en la flecha central (del medio). Si la flecha del medio apunta hacia la izquierda, presiona la tecla shift izquierda, si la flecha del medio apunta hacia la derecha, presiona la tecla shift derecha. Procura hacerlo lo más rápido posible. Recuerda que lo importante es que te concentres en la flecha central.

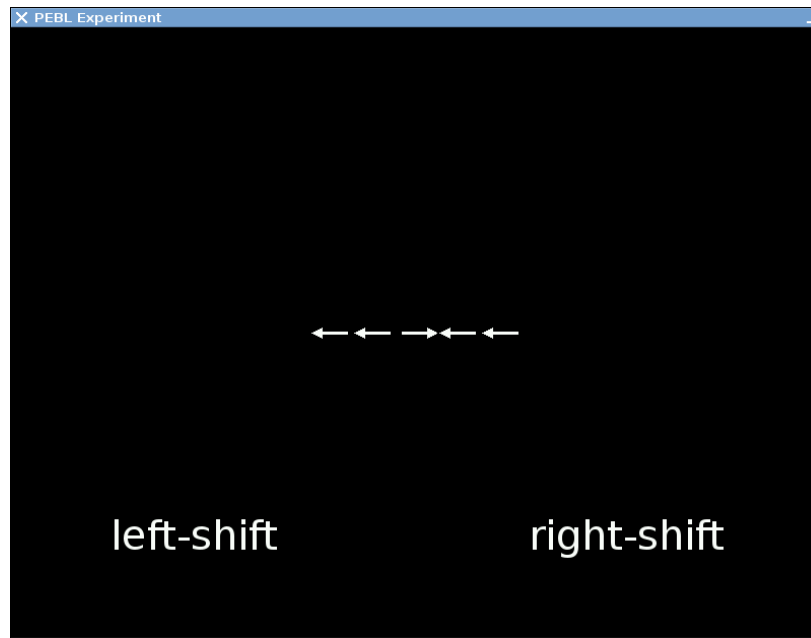
Si has comprendido la consigna, presiona la barra espaciadora para continuar.



De acuerdo al mapping anterior la persona en este caso deberá presionar la tecla shift derecha. Ahora veamos un mapping diferente con los mismos estímulos:

A continuación se te presentará una serie de flechas (cinco). Debes concentrarte en la flecha central (del medio). Si la flecha del medio apunta hacia la izquierda, presiona la tecla shift izquierda, si la flecha del medio apunta hacia la derecha, presiona la tecla shift derecha. Aparecerán a los costados de la flecha central flechas distractoras, que pueden o no tener la misma dirección que la flecha central. Debes ignorar las flechas distractoras, y concentrarte solo en la flecha central. Procura hacerlo lo más rápido posible. Recuerda que lo importante es que te concentres en la flecha central, ignorando las distractoras.

Si has comprendido la consigna, presiona la barra espaciadora para continuar.



La presentación estimular es la misma, sin embargo, la persona recibió en el mapping la orden de ignorar las flechas que hacen de flanco de la flecha central. Esto activa el control inhibitorio, la persona estará preparada de antemano para desarrollar estrategias de control perceptual y asignará recursos a priori. Es probable que con este nuevo mapping los primeros ensayos resulten más rápidos que si se hubiera utilizado el primer mapping (esto podría ser sometido a prueba por parte del lector- programador).

Ahora, veamos una variación mucho más radical del mapping sobre la misma presentación estimular:

A continuación se te presentará una serie de flechas (cinco). Debes concentrarte en las flechas de los extremos izquierdo y derecho, ignorando la flecha central (del medio). Si las flechas de los costados apuntan hacia la izquierda, presiona la tecla shift izquierda, si las flechas de los costados apuntan hacia la derecha, presiona la tecla shift derecha. Procura hacerlo lo más rápido posible. Recuerda que lo importante es que te concentres en las flecha de los costados y que ignores la central.

Si has comprendido la consigna, presiona la barra espaciadora para continuar.

Este último mapping cambia el diagrama de flujo que elicit la situación conductual. Al pedirle a la persona que responda en dirección a lo que antes se concebía como distractores, disminuye notablemente el efecto de interferencia. La persona deberá recurrir mucho menos al control

ejecutivo atencional, y más a la rapidez de procesamiento visual. La flecha central provoca una interferencia menor que los flancos, y la respuesta estará mediatizada por la velocidad de procesamiento de la información.

La habilidad para construir (redactar) mappings se vincula con la claridad para organizar comportamientos y es importante desarrollarla. La programación del mapping se relaciona con la asignación de tiempos específicos (por ejemplo 15.000 milisegundos por permanencia de pantalla) o relacionados al evento que significa que la instrucción desaparece solo cuando la persona cree que comprendió la tarea. La combinación entre presentaciones estimulares, condiciones aplicadas sobre presentaciones estimulares y mapping configuran la situación conductual y un diagrama de flujo acompañará conceptualmente estas relaciones.

5. Tipos de diseños de programación de experimentos

Se aprecia que en la programación de experimentos hay eventos que se repiten, los estímulos son las 'estrellas' de la programación ya que la persona 'afrontará' su estructura en su desempeño conductual. Se ha mencionado que podrían ser videos, audios, palabras, imágenes cualquiera y combinaciones entre estos elementos. Se puede presentar lo que el investigador desee, lo importante es que la programación permite hacerlo de forma organizada y controlada. Se entiende por organizada que se asignan tiempos estables de presentación, se asigna una lógica de presentación de secuencias estimulares y el mapping organiza que las secuencias se relacionen con un modelo cognitivo. Se obtendrán respuestas que informarán sobre cómo respondió cada persona en accuracy y tiempos de reacción.

Se repasarán diferentes tipos de diseños en programación experimental, que representan modos diferentes de presentar los eventos, creación de condiciones sobre los estímulos y obtención de respuestas.

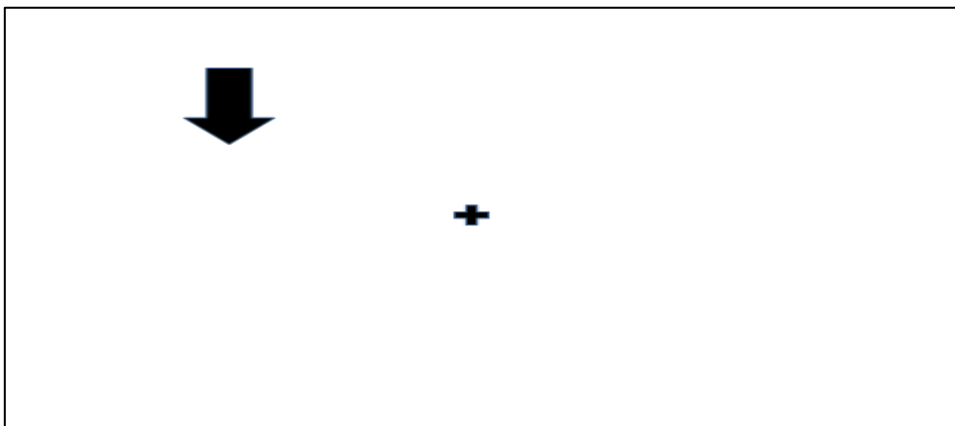
Programación para la organización de estímulos: son programaciones que objetivan los tiempos de presentación de estímulos prolongados y pueden estar seguidos de escalas de evaluación. Por ejemplo, un investigador que pasa un video de dos minutos de duración y que inunda la emoción de tristeza. Se programa el video para que dure exactamente ese tiempo y luego la persona responde a un cuestionario también programado en ordenador. Se trata en este ejemplo de un cuestionario conocido, como la escala de depresión de Beck. Luego de completarlo, presionando teclas que actúan como opciones de respuesta ante cada pregunta o pedido de información, la persona vuelve a visualizar un video de dos minutos de duración que inunda la emoción de alegría y luego vuelve a completar el mismo inventario de depresión.

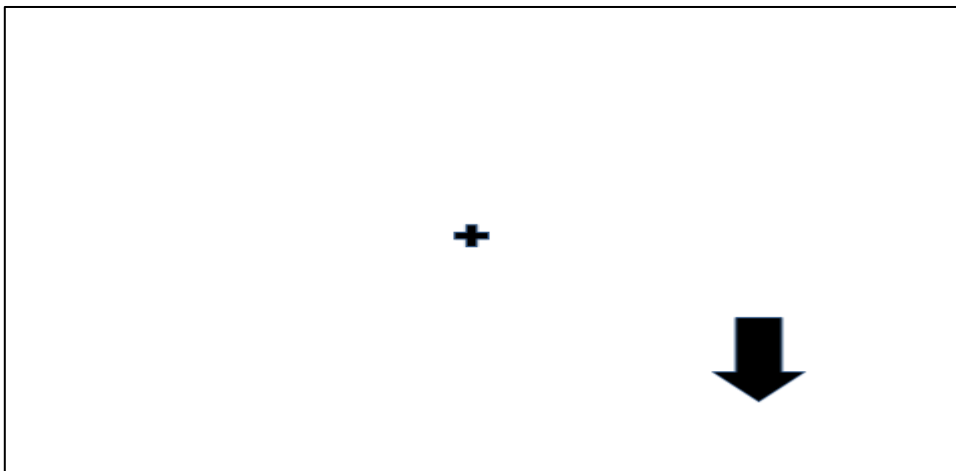
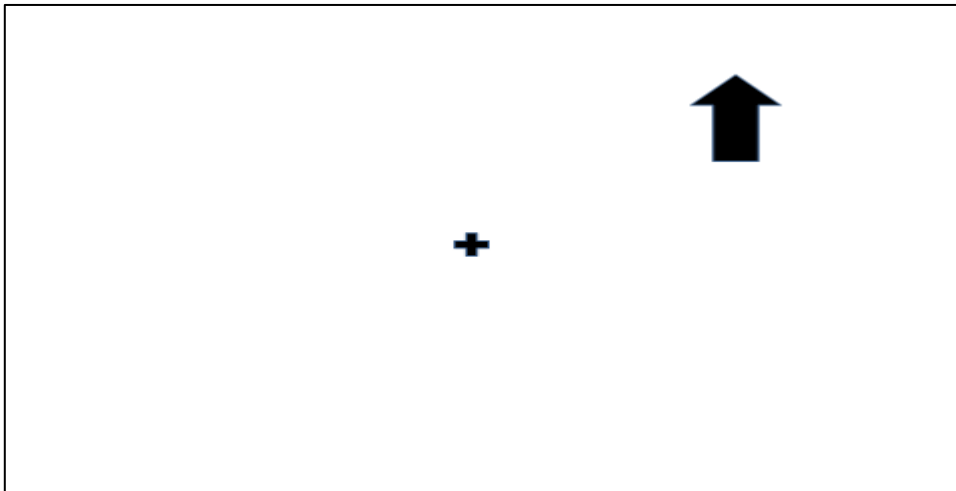
En este ejemplo, la programación sirvió para coordinar la presentación de los estímulos. Los eventos fueron: un video de tristeza, 40 preguntas- ítems del inventario de depresión, un video de alegría, 40 preguntas – ítems otra vez. Durante la presentación de los videos el mapping indicaba que solo debía visualizarlos, intentando no alterar su respuesta facial (supresión). El mapping frente a los inventarios indicaba leer el reactivo y responder a la opción que mejor describía lo que le sucedía en su vida personal y que mejor describía su comportamiento habitual. En este ejemplo,

los tiempos de reacción no son tenidos en cuenta y se aprenderá en la parte 2 como indicarle al programa que no colecciona tiempos de reacción. Durante la visión de los videos la persona no debe realizar respuesta alguna, en cambio durante los inventarios se registrará qué respuesta concreta emitió la persona (por ejemplo qué número presionó) y no habrá respuestas correctas o incorrectas. En la parte 2 se aprenderá a habilitar y deshabilitar teclas de respuestas en tanto correctas o indiferentes a una calificación de ese tipo.

Programación cross-tasking: esta es una de las presentaciones más complejas y eficaces. Se caracterizan por utilizar una estructura (matriz) estimular que admite variaciones concretas. Por ejemplo, se presentan rostros y marcos de colores (matriz estimular) pero los rostros y colores varían (concretización diferencial). Las variaciones diferenciales responden a diferentes condiciones, pero se mantiene la estructura estimular. Esto permite que los eventos sean estructuralmente similares entre sí, pero las variaciones por concretizaciones posibiliten la comparación entre condiciones. Son diseños que remiten a un análisis intra- sujeto, ya que las condiciones de interés se comparan dentro de una misma persona, y esto reduce numerosos factores de azar, los que aparecerían si se utilizaran dos grupos diferentes de personas realizando la misma tarea y luego se comparan. Se brindará un ejemplo de programación cross- tasking, aunque los ejemplos anteriores de experimentos en su mayoría así lo han sido. Una de las ventajas de esta programación es que permite reducir la cantidad de participantes por experimento.

Se pide durante el mapping que las personas respondan acerca de la dirección a la que apuntan las flechas. Si apuntan hacia arriba, deben presionar la tecla C y si la flecha apunta hacia abajo deben presionar la tecla M, lo más rápido posible. La persona debe fijar su mirada en una cruz central que aparece en el centro de la pantalla (punto de fijación). Los estímulos son flechas que apuntan hacia arriba o hacia abajo, y la pantalla está dividida en cuatro cuadrantes: superior izquierdo, superior derecho, inferior izquierdo, inferior derecho. En el teclado QWERTY se aprecia que la tecla C está a la izquierda de la tecla M, las dos teclas de respuesta. Esta combinación mapping- estímulos ha creado dos conflictos perceptuales y cuatro condiciones. Se verán en primer lugar tres ejemplos de eventos con esta organización estimular:

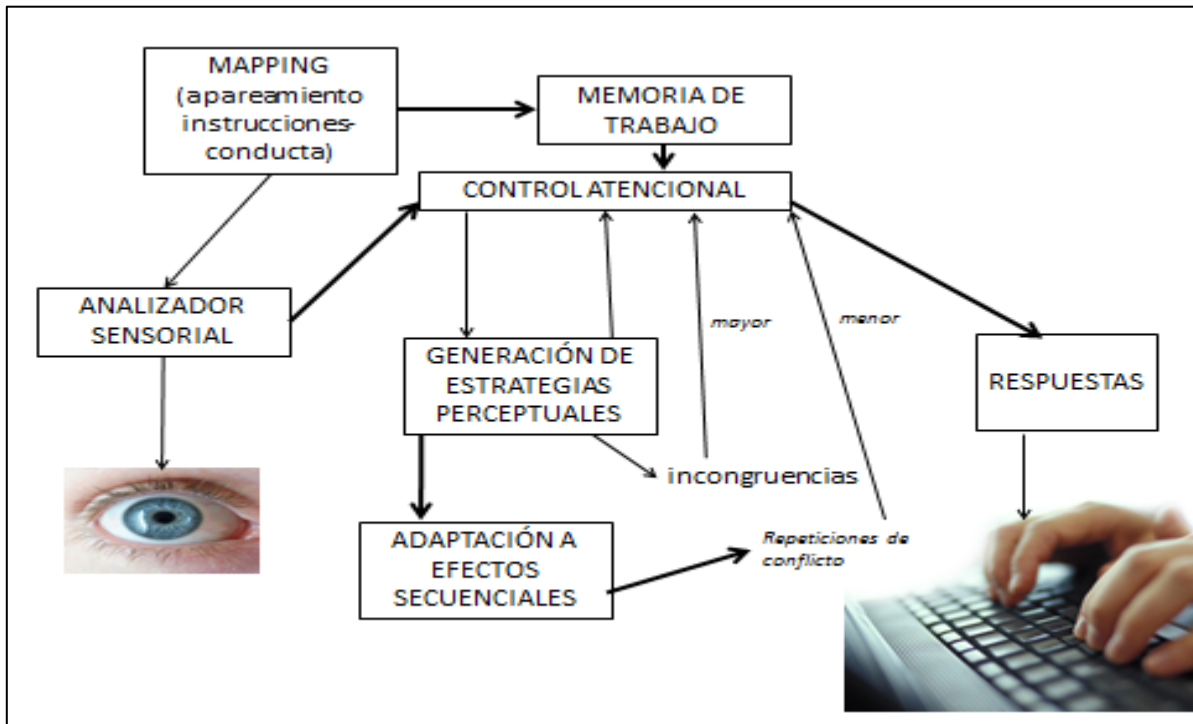




El conflicto perceptual Stroop se define como un conflicto estímulo- estímulo, en este caso la flecha aporta información direccional (señala hacia arriba o hacia abajo) y aparece en un hemisferio espacial (el de arriba o el de abajo). Hay un Stroop congruente cuando la flecha que apunta hacia arriba aparece en el hemisferio de arriba y también cuando la flecha apunta hacia abajo y aparece en el hemisferio de abajo. El Stroop incongruente aparece cuando hay un cruce de ambas informaciones, por ejemplo la flecha apunta hacia abajo y está en el hemisferio de arriba.

El conflicto perceptual Simon se define como uno de naturaleza estímulo- respuesta. El efecto Simon indica que se es más rápido y eficaz respondiendo hacia la misma orientación espacial de la que partió la información estimular. En este caso, cuando una flecha que apunta hacia arriba aparece en el hemisferio izquierdo habrá congruencia Simon, porque la tecla de respuesta está a la izquierda (y requiere responder con la mano izquierda) y cuando la flecha que apunta hacia abajo aparece a la derecha también habrá congruencia Simon por la misma razón pero en referencia a la mano derecha. Habrá incongruencia Simon cuando las informaciones cuando la flecha que apunta hacia abajo aparezca en el hemisferio izquierdo y viceversa.

Este diseño es cross- tasking porque con la misma presentación estimular se presentan las cuatro condiciones. En el evento 1 de este experimento hay incongruencia Stroop incongruencia Simon, en el evento 2 hay congruencia Stroop incongruencia Simon, en el evento 3 hay congruencia Stroop congruencia Simon. Como quedaron definidas cuatro condiciones, si consideramos el número mágico 30 eventos por condición, se aprecia que con 120 eventos sería un buen experimento. Se presenta el diagrama de flujo de este experimento:



Programación de eventos simples: en este caso se presentan eventos sucesivos implicando cada evento una sola condición. A diferencia del cross- tasking, que en cada evento se superponían condiciones, aquí cada evento representa una sola condición, como el caso del experimento de los círculos rojos que se desplazaban entre círculos en blanco en la dirección de las agujas del reloj. Cada evento era un cambio de la posición del círculo rojo, había dos tipos de condiciones en el experimento, inhibición de respuestas cuando el círculo rojo se desplazaba al círculo vacío consecutivo y vigilancia cuando el círculo rojo saltaba un círculo y la persona debía presionar la barra espaciadora. Cada evento suponía que se presentaba o bien una condición o bien la otra.

Programación de eventos secuenciales: Son experimentos donde las condiciones emergen de la relación entre un evento y el siguiente. Por ejemplo, recordemos el experimento de las fotografías de varones y mujeres que tenían diferentes edades. Estaban enmarcadas por colores que determinaban si debía responderse por una u otra condición. Una de las condiciones es el cambio de sexo, otra condición el cambio de edad y otra el cambio de consigna. En este caso se podía encontrar que en la sucesión de eventos había un cambio de sexo, no cambio de edad y cambio de marco. El interés conceptual de este experimento residía en su activación de recursos de flexibilidad cognitiva. Se aprecia que las condiciones emergen de la sucesión de eventos, de la

relación entre un evento y el que le sigue. A diferencia del cross tasking, donde las condiciones de interés estaban en el mismo evento, superpuestas, aquí las condiciones se activan en la sucesión, aunque en los diseños cross tasking también es muy frecuente que se activen condiciones secuenciales.

6. ¿Qué es un atributo?

Varios conceptos muy relacionados y cercanos han ido apareciendo en esta exposición: estímulo, condición, evento, diseño. Corresponden a diferentes niveles de análisis: el estímulo tiene una concretud perceptual, provoca procesamiento de la información, es una porción objetivada de realidad que manipula el investigador en la situación conductual. El evento corresponde a la organización temporal del experimento, y es la unidad mínima de tiempo en la organización de la presentación estimular, suele representar un pulso de información en la interface de software que combina experimentos programados con adquisiciones fisiológicas. Una condición se corresponde con un nivel conceptual y se relaciona con los conocimientos que tiene el investigador de los procesos cognitivos y qué variables desea investigar. Corresponde a un nivel de análisis metodológico- conceptual. El diseño se vincula al programa en general y tiene consecuencias en los análisis estadísticos que se harán sobre los datos que arroja el experimento.

Un atributo representa una propiedad del estímulo, algunos son concretos y otros son conceptuales. En los capítulos posteriores se aprenderá a programar mediante la vinculación lógica de los atributos, dentro de un objeto clave en programación denominado 'List', pero ahora se hará un análisis más teórico utilizando ejemplos. Veamos los estímulos del primer experimento:



1.bmp



2.bmp

Los atributos deconstruyen el estímulo en sus componentes básicos, transformando cada concretización (los dos rostros enmarcados que se ven arriba son concretizaciones) en una estructura de atributos relacionados. La estructura estimular implica las categorías genéricas que lo componen, en estos casos se tiene Sexo, Edad, Marco y queda conformada la siguiente tabla:

FOTOGRAFÍA	SEXO	EDAD	MARCO
1.bmp	Mujer	Joven	verde
2.bmp	varón	Avanzada	morado

Sin dificultad, se podría hacer el siguiente cambio:

FOTOGRAFÍA	SEXO	EDAD	MARCO
1.bmp	Mujer	Joven	morado
2.bmp	varón	Avanzada	verde

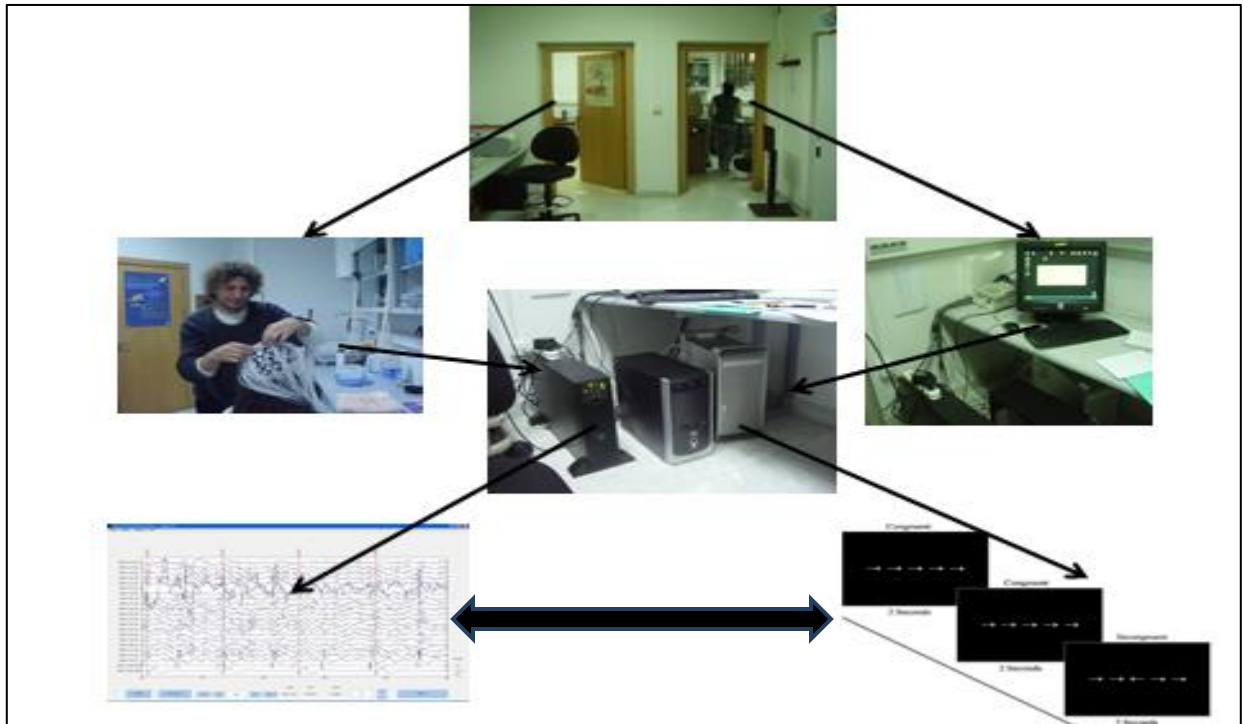
Y obtener la siguiente tabla:

FOTOGRAFÍA	SEXO	EDAD	MARCO
1.bmp	Mujer	Joven	verde
2.bmp	varón	Avanzada	morado
1.bmp	Mujer	Joven	morado
2.bmp	varón	Avanzada	verde

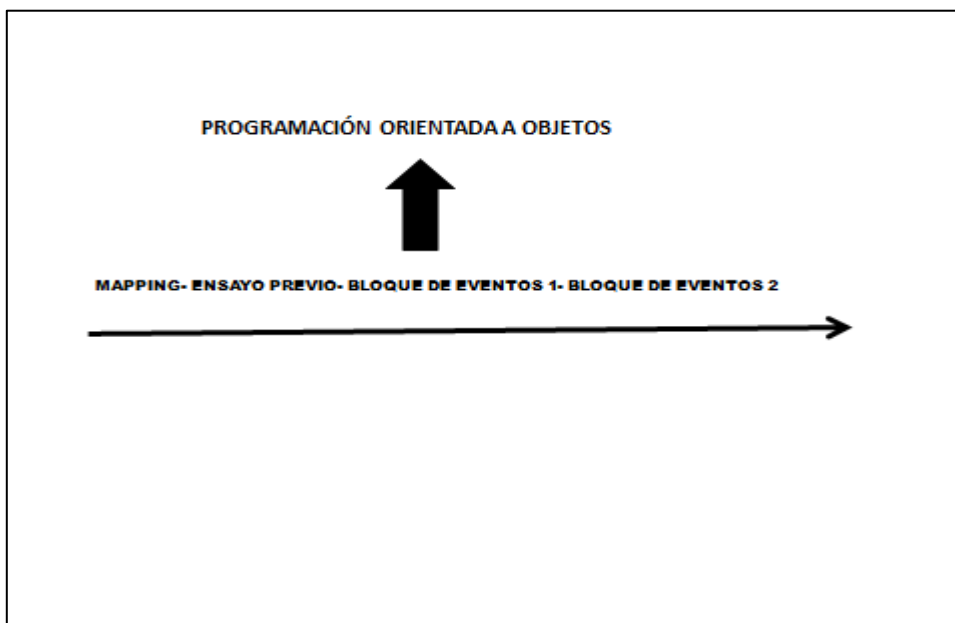
Se aprecia que el atributo FOTOGRAFÍA apunta a una imagen concreta, un archivo que debe estar presente en el ordenador, en la misma carpeta donde se guarda el 'script' o guión del experimento que se está programando. En la parte 2 se detallará que es un script y de qué modo se facilita su escritura mediante la programación orientada a objetos. Los atributos SEXO y EDAD se aprecia que son descriptivos de la fotografía concreta, sirven para ir calculando cuantas veces se presenta cada uno. Sin embargo el atributo MARCO representa una combinación, ya que no es intrínseco a la concretización del atributo FOTOGRAFÍA sino que será resultado de una superposición y de este modo, el marco puede tener diferentes colores en una misma concretización (fotografía). El objeto 'slide' que será desarrollado en la parte 2 podrá ser muy útil para realizar combinaciones de atributos y presentarlas en un estímulo.

7. La programación de experimentos en Psicología es una línea de tiempo

Con el transcurso de los ejemplos cada vez irá quedando más claro que se programará sobre la base de una línea de tiempo, indicando al ordenador qué objetos se irán haciendo presentes, cuantos milisegundos permanecerá cada uno y cuantos eventos serán. El evento es el elemento más repetitivo de la programación, ya que se aprovecha de que la estructura estimular y sus atributos son combinables para presentarse varias veces. Un experimento puede componerse por ejemplo de 500 eventos, lo cual resulta habitual cuando el experimento recoge datos conductuales de forma simultánea con la recogida de datos neurofisiológicos, por ejemplo potenciales evocados.

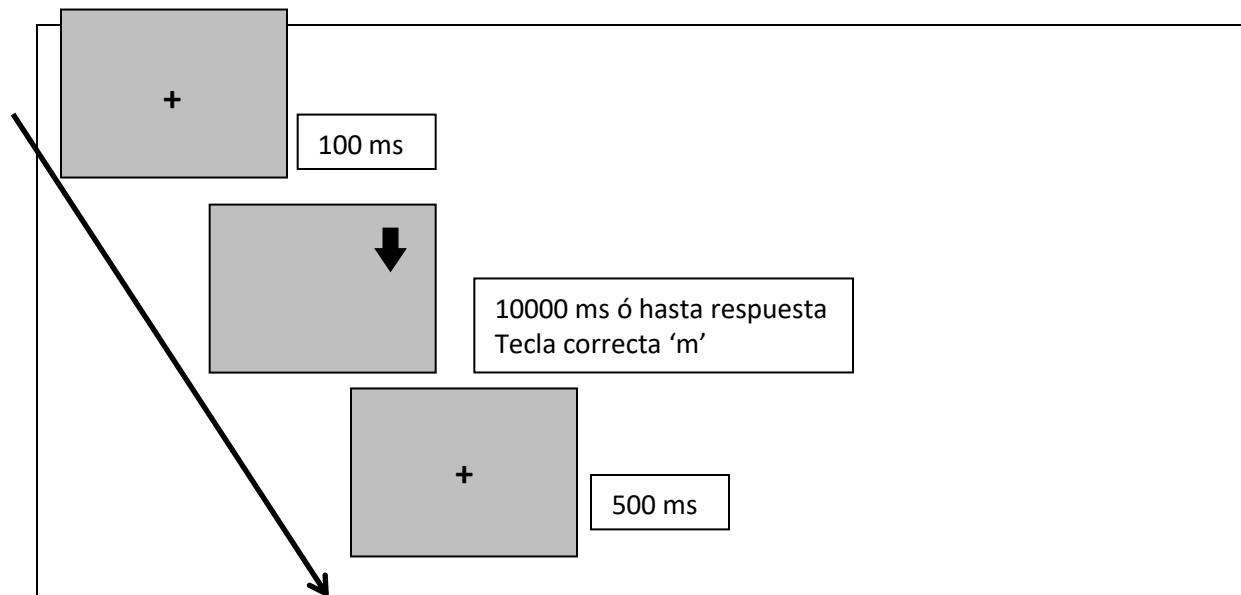


En la parte 2 se aprenderá a colocar los objetos de forma adecuada en la línea de tiempo, habrá objetos muy vinculados con la presentación del mapping, como los textos, objetos que representan la concretización estimular, como los 'slides', objetos para establecer las posibles respuestas conductuales, objetos relacionados con el feedback y objetos para organizar la lógica de pasaje de los eventos, entre otros. También se verá que resultan importantes los conceptos de presentación aleatoria, contrabalanceo y presentación por bloques de eventos.

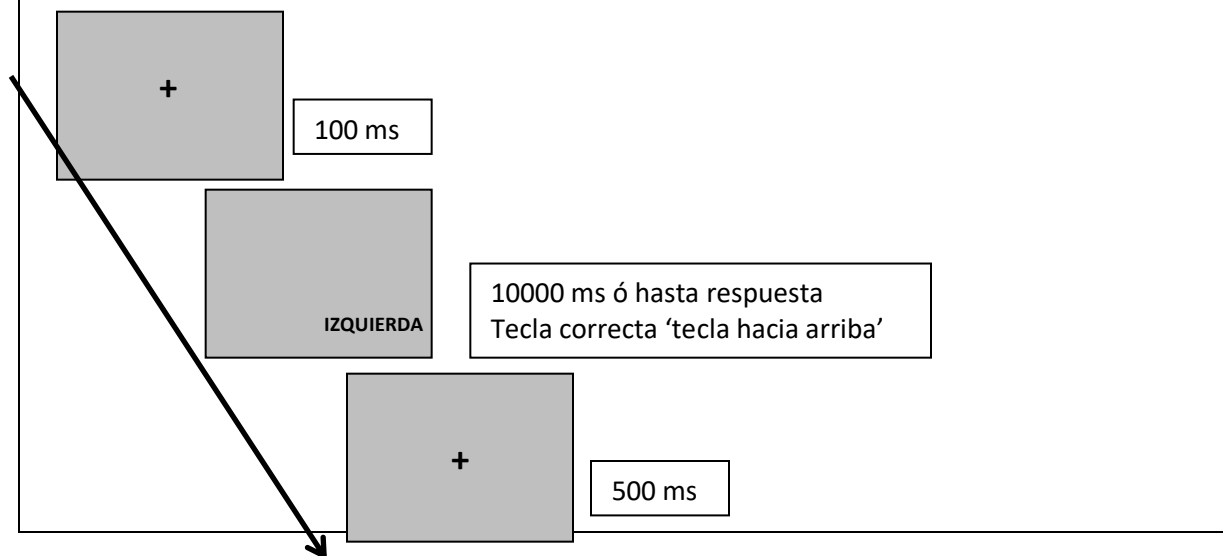


En los artículos de investigación ('papers') es muy frecuente que la temporalidad de los experimentos programados se represente de la siguiente manera:

Efecto Stroop, sin efecto Simon, con estímulo iconográfico



Efecto Stroop con efecto Simon y presentación semántica



8. Crítica a los diagramas de flujo como analogías del flujo de información: los modelos de propagación en paralelo

Hay que considerar una fuerte crítica a los diagramas de flujo secuenciales en Psicología Cognitiva y a su interpretación teórica. Se ha postulado que implican que habría un procesamiento secuencial y que la información circularía a través de módulos. Sin embargo, concepciones

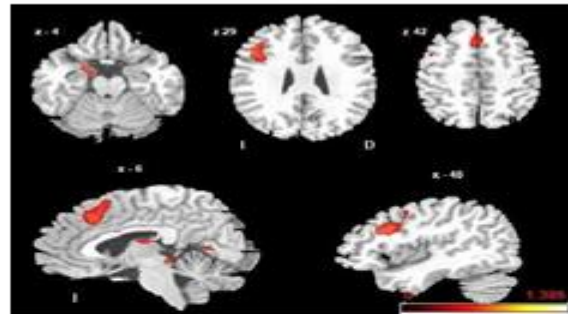
actuales en neurociencias (y no tan actuales, como los modelos PDP de los años '80) sostienen que la actividad nerviosa procedería en paralelo con activaciones simultáneas en centros nerviosos diferentes. De esta forma no se requeriría de modelos secuenciales de flujo análogo de información, sino de modelos teóricos que establezcan claramente qué variables están implicadas en la tarea programada y qué relaciones lógicas tienen entre sí.

Gráficamente puede no haber muchas diferencias, las variables admiten una relación de variables independiente- dependiente, de covariación o de mediación entre sí. A la vez, se describen diferentes estratos de variables, por lo que una variable podría ser dependiente de una independiente y a la vez independiente de una nueva variable que a su vez es variable independiente de otra situada en un nuevo estrato. La cuestión clave reside en si los diagramas de flujo cognitivos son analogías del flujo de información en el sistema cognitivo o si representan modelizaciones que no se postulan como mecanismos de instanciación de estos procesos. Son cuestiones epistemológicas que también exceden el alcance de esta guía, pero conviene mencionarlás porque se sugiere que la programación de un experimento en Psicología debe tener un respaldo conceptual que generalmente se representa como modelo, y es frecuente indagar acerca de cómo debe interpretarse la relación direccional entre las cajas que componen las modelizaciones en Psicología.

La aceptación de los cuestionamientos a la interpretación secuencial y analógica de los diagramas de flujo cognitivos está acompañada de avances en los conocimientos del funcionamiento de las redes neuronales. Implicaría que una tarea programada en un experimento supone la activación específica de un proceso o mecanismo cognitivo, el experimento o tarea sería la presentación organizada de una estimulación que los pone en actividad. Por ejemplo la tarea ANT- IV presenta flancos, asteriscos, señales de alertas auditivas y sutiles desviaciones perceptuales en la construcción de los estímulos de vigilancia que por su complejidad activarían las redes neuronales de orientación, alerta, control y vigilancia, representadas por redes neuronales que aumentan su actividad cuando se les presenta una estimulación con esas características.

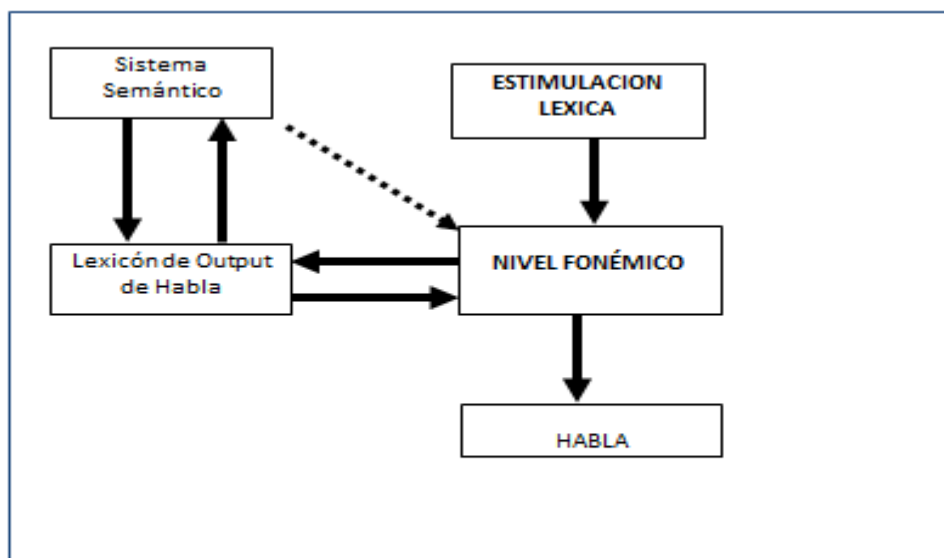
Esto no supone que sigan un flujo representable por módulos secuenciales, sino que ciertas regiones de sustancia gris conectadas por fibras específicas de sustancia blanca comienzan a aumentar o inhibir significativamente su actividad, en comparación con una 'default network' o 'red de funcionamiento cognitivo por defecto'. La potencia de esta interpretación reside en que la tarea programada representaría conductualmente un proceso cognitivo específico y se necesita para su validación que existan datos de neuroimágenes y fisiológicos obtenidos en simultaneidad con la recogida de datos conductuales. También se destaca que esta concepción no supone un abandono de los conceptos de capacidades cognitivas, funciones ejecutivas y modalidades sensoriales, sino de un cuestionamiento de que sus interacciones se puedan representar adecuadamente mediante flechas secuenciales. Los mismos diagramas presentados significarían módulos en interacción constante, unos más conectados entre sí que otros, y el flujo no seguiría una dirección entrada- salida sino que habría una emisión constante de parte de cada uno, que se podría modelizar más adecuadamente con algoritmos de redes neuronales.

En las tareas de fluidez verbal, donde una persona evoca palabras pertenecientes a una determinada categoría o criterio específico, se encontró mediante datos de neuroimágenes la siguiente activación cerebral (datos obtenidos por nuestro grupo de investigación) (Marino Dávalos, Redondo, Luna, Sánchez, & Foa Torres, 2012):



Datos de neuroimagen obtenidos mediante resonancia magnética funcional cuando una persona evocaba verbos

Y un modelo cognitivo de la evocación de verbos (la misma prueba de fluidez verbal cuya actividad cerebral se muestra arriba) fue representado mediante un diagrama secuencial de flujo:



Si se comparan las dos fuentes de datos, actividad cerebral cuando se evocan verbos en resonancia magnética funcional y diagrama de flujos secuencial cognitivo ¿llegaríamos a la misma conclusión sobre la actividad cognitiva involucrada? Un buen método para abordar estas cuestiones consiste en la utilización combinada de datos de neuroimágenes interpretados con algoritmos de ontologías cognitivas y datos conductuales analizados mediante técnicas de modelización de ecuaciones estructurales. La explicación de este abordaje excede el alcance de esta guía, pero es una dirección que se sugiere y se ha emprendido como aporte empírico y con evidencias ante un debate que fácilmente podría ‘empantanarse’ en las movedizas arenas de la epistemología especulativa.

9. Preparación para el aprendizaje operativo de programación de experimentos en Psicología

A continuación se presentará un ‘checklist’ a la manera en que los pilotos de aviones repasan la adecuación de los dispositivos de la aeronave antes del despegue. Se sugiere que antes de iniciar el diseño de un experimento se repase este ‘checklist’, que también servirá de orientación para iniciar la fase ‘operativa’ de programación. El estudiante debe comprender que cada ítem incorporado a esta lista es uno que debe aprender a dominar como ‘programador’.

CHECKLIST PARA PROGRAMAR EXPERIMENTOS EN PSICOLOGÍA:

- 1- Tener en claro el concepto que está 'detrás' del experimento.
- 2- Repasar los modelos teóricos existentes sobre el concepto que se activará mediante el experimento.
- 3- ¿Qué deseo activar del sistema cognitivo de la persona?
- 4- ¿Qué antecedentes conozco que provoquen una activación parecida?
- 5- ¿Qué tipo de estímulos van a ser presentados? ¿Son imágenes, sonidos, diapositivas, palabras, videos? ¿Algún otro?
- 6- ¿Se van a presentar diferentes condiciones en la presentación estimular? ¿Cuáles?
- 7- ¿Qué atributos componen los estímulos?
- 8- ¿Qué tipos de respuesta quiero recoger?
- 9- ¿Cómo haré para que las instrucciones sean comprendidas con claridad?
- 10- ¿El mapping será sólo lectura de instrucciones o tendrá mi propia ayuda?
- 11- ¿Necesito presentar un bloque de ensayos previos para asegurar que se comprendió la consigna?
- 12- La presentación: ¿Implica diferentes bloques?
- 13- ¿Debería darle feedback a las personas? Si debería hacerlo ¿Será durante todo el proceso?
- 14- ¿Cuántos eventos quisiera presentar?
- 15- La cantidad de eventos: ¿Resulta lógica en relación a los atributos de los estímulos?
- 16- Los eventos: ¿Tienen un tiempo límite de presentación?
- 17- ¿La presentación de los eventos será aleatoria?
- 18- ¿Es necesario presentar un tipo de contrabalanceo dentro del experimento?
- 19- ¿Cuál es la estructura de los estímulos que se van a presentar?
- 20- ¿Qué atributos explícitos conforman la estructura de los estímulos?
- 21- ¿Cuáles son los atributos implícitos que tendría que incluir para mejorar el análisis de datos?
- 22- ¿Qué tipo de diseño de programación es el más adecuado?
- 23- ¿Por qué una persona sería seleccionada para participar del experimento?
- 24- Las personas participantes: ¿Serán expuestas a tratamientos que se supone tendrán efectos en el rendimiento en el experimento?
- 25- ¿Hay otras variables que debería medir para aumentar la validez del experimento?
- 26- ¿Cómo se les pedirá el 'consentimiento informado'?
- 27- ¿Hay alguna cuestión ética que deba tener especialmente en cuenta?

REFERENCIAS PARTE 1:

- Badre, D., & Wagner, A. D. (2007). Left ventrolateral prefrontal cortex and the cognitive control of memory. *Neuropsychologia*, 45(13), 2883–2901.
- Botvinick, M. M., Braver, T. S., Barch, D. M., Carter, C. S., & Cohen, J. D. (2001). Conflict monitoring and cognitive control. *Psychological review*, 108(3), 624.

- Callejas, A., Lupiáñez, J., & Tudela, P. (2004). The three attentional networks: On their independence and interactions. *Brain and cognition*, 54(3), 225–227.
- Crone, E. A., Richard Ridderinkhof, K., Worm, M., Somsen, R. J. M., & Van Der Molen, M. W. (2004). Switching between spatial stimulus–response mappings: A developmental study of cognitive flexibility. *Developmental Science*, 7(4), 443–455.
- Marino Dávalos, J., Redondo, S., Luna, F. G., Sánchez, L. M., & Foa Torres, G. (2012). Actividad cerebral medida con Resonancia Magnética Funcional durante la prueba de fluidez de acciones. *Revista Neuropsicología Latinoamericana*, 4(4), 28–35.
- Mitchell, K. J., Johnson, M. K., Raye, C. L., & D’Esposito, M. (2000). fMRI evidence of age-related hippocampal dysfunction in feature binding in working memory. *Cognitive brain research*, 10(1), 197–206.
- Miyake, A., Friedman, N. P., Emerson, M. J., Witzki, A. H., Howerter, A., & Wager, T. D. (2000). The unity and diversity of executive functions and their contributions to complex ‘Frontal Lobe’ tasks: a latent variable analysis. *Cognitive Psychology*, 41(1), 49–100.
doi:10.1006/cogp.1999.0734
- Ochsner, K. N., & Gross, J. J. (2005). The cognitive control of emotion. *Trends in Cognitive Sciences*, 9(5), 242–249. Retrieved from <http://www.ncbi.nlm.nih.gov/pubmed/15866151>
- Posner, M. I., & Petersen, S. E. (1989). *The attention system of the human brain*. DTIC Document.
- Rumelhart, D. E., & McClelland, J. L. (1987). *Parallel Distributed Processing: Explorations in the Microstructure of Cognition. Foundations* (Vol. 1). MIT press.
- Vega Rodríguez, M. de. (1984). *Introducción a la psicología cognitiva*. Alianza Editorial.

PARTE 2: ASPECTOS OPERATIVOS DE LA PROGRAMACIÓN DE EXPERIMENTOS EN PSICOLOGÍA

1. Inicio de la programación

Esta guía está dirigida a psicólogos y estudiantes de psicología, que habitualmente no están familiarizados con la programación. El funcionamiento del lenguaje de programación es lógica en acción, y memoria de la utilidad de cada objeto, es una combinación de memoria y lógica por lo que, nuevamente, se invita a la regulación de emociones y a ponerlas al servicio de la recuperación de contenidos y del esfuerzo para construir edificios lógicos complejos y atractivos. E- Basic (EB) es amigable y facilita la programación de modo que gran parte de la lógica que se aplicará está centrada en el manejo del objeto 'List' y de la memoria de los 'nombres' 'etiquetas' que se van incluyendo en los atributos y objetos que 'List' controla. Se programará en una línea de tiempo, que es el primer objeto que se arrastrará a la consola. El objeto 'List' es la cabina de comando del avión, desde allí los objetos que se presentarán son controlados, nominados, dirigidos. Los objetos que son comandados por 'List' son variados y se corresponden en su mayoría a lo que visualizará el participante. Y luego hay otros objetos que se añaden y parámetros que se asignan que exceden a 'List' y a los objetos que controla 'List', pero que influyen en la línea de tiempo. Progresivamente, con la maestría de dominio, aparecerá el interés por la inclusión de 'InLines' porciones de código escrito que superan las prestaciones de los objetos que se arrastran. Eso quedará para más adelante, cuando se haya cristalizado la programación orientada a objetos con los elementos basales que la constituyen.

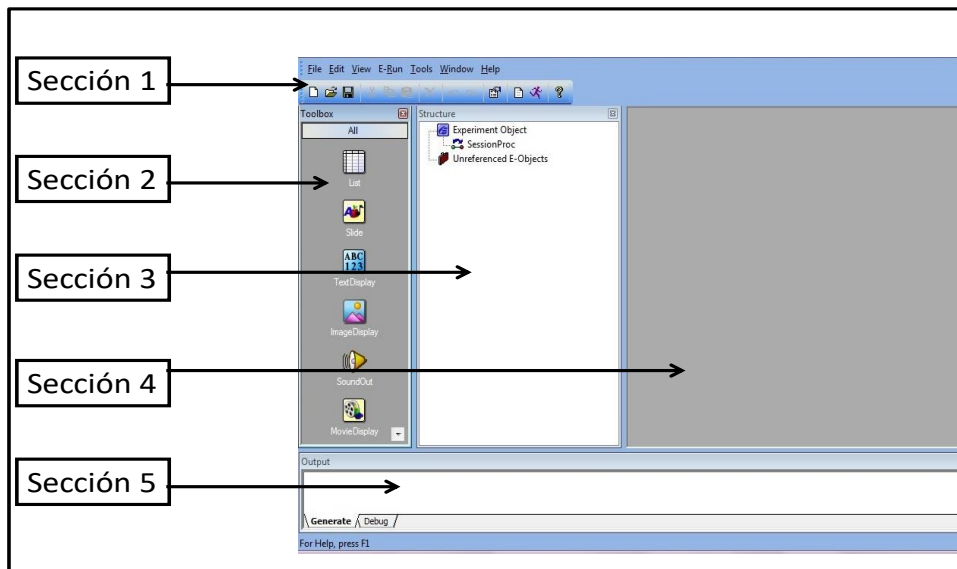
Cuando la persona se sienta a programar debe tener en claro qué desea realizar. En un papel al costado de su ordenador debería trazar una línea de tiempo y definir claramente qué quiere presentar a los participantes, incluida la configuración específica que tendrán los estímulos. Hay que evitar 'sentarse' sin una meta clara, eso implicaría 'vagabundear' por el programa sin poder discriminar luego qué se hizo bien y qué se hizo mal, cuestiones que solo aparecen si está en claro que se deseaba lograr. Ante un eventual error el programador podrá fragmentar en que paso ha ocurrido, y sería idóneo llevar una libreta de apuntes de errores, para luego evitar volver a cometerlos y aprender cómo debía escribir cualquier parte del script. El programador también debería acostumbrarse a trabajar siguiendo 'pipelines' que son algoritmos o pasos sucesivos que tienen un estado inicial y un estado final y movimientos. Se suelen representar mediante una lista de acciones consecutivas, se van ejecutando y se logra un estado que se compara con el deseado. Un pipeline exitoso se guarda en formato electrónico, se le coloca un título descriptivo (por ejemplo Pipeline 1: 'Inclusión de inlines para modular el nivel de exigencia de acuerdo al rendimiento') y de esa manera se automatizan las diferentes partes del proceso de programación. Hay que evitar que las dificultades que aparecen cuando se inicia un nuevo proceso vuelvan a aparecer cuando se quiere programar un proceso similar en otro experimento. Progresivamente, se irá conformando una 'biblioteca' de pipelines dotados de títulos claros para que el programador pueda encontrar lo que busca.

La parte 2 se desarrollará con ejemplificaciones del lenguaje e-basic (EB), que es una 'acomodación' de Visual Basic (VB) adaptada a los objetivos de experimentos en psicología. Es un lenguaje de Alto Nivel, que significa que requiere de compiladores que 'traduzcan' las instrucciones (programación orientada a objetos) a lenguaje de máquina (bajo nivel). EB funciona por supuesto de un modo lógico, hace lo que se le indicó que haga, aclaración que aparenta obviedad y redundancia, sin embargo, el estudiante de psicología poco acostumbrado a estas manipulaciones puede frustrarse cuando espera que 'las cosas' sucedan de un modo, y aparecen errores o suceden de modo inesperado. Entonces el programador debe apelar a la regulación de sus emociones, en especial de la frustración, y tener en cuenta que lo aconteciendo corresponde a órdenes que pueden ser explícitas (el programador de forma intencional arrastró un objeto, estableció un parámetro) o implícitas (EB asume valores por defecto). Esto significa que a veces se realizan acciones que no fueron ordenadas, pero EB asignó un valor 'de relleno', 'empaquetado', 'de programa'. En los inicios es probable que aparezcan varios errores, para superarlos y aprender de ellos es vuelve a resaltar actuar según está disciplina: Trazar una meta, intentar plasmarla recuperando los conocimientos que se tienen, observar el resultado, registrar los errores y discordancias con lo esperado, leer qué tipo de error se ha presentado, iniciar una reprogramación. Se registra el error, qué se hizo para cometerlo (o qué 'no se hizo') y se pasa a pipeline el resultado exitoso. Se empaqueta el pipeline guardándolo en archivo y colocándole un título que describa para qué sirve y se integra a la biblioteca de recursos. A medida que el programador se va volviendo experto, la cantidad de parámetros que se asignan por defecto va disminuyendo, ya irá paulatinamente 'conquistando' cada paso y objeto que conforma el 'script'.

2. Interfaz del software

Una vez abierto EB, existen dos modos de comenzar a programar: mediante una plantilla Basic (en español, básica) o mediante una plantilla 'Blank' (en español, en blanco). Se debe seleccionar y aceptar el modo 'Blank' de programación. Desde ambas plantillas se programará mediante el arrastre de objetos y la manipulación de sus propiedades (conocido como Programación Orientada a Objetos -POO-). Si se selecciona la plantilla básica, EB provee una estructura básica de un experimento para ser modificado y programado. El objetivo será programar un experimento desde el inicio, por ese motivo se selecciona la plantilla en blanco.

La interfaz que se presenta al usuario a continuación puede ser visualizada en la siguiente figura. En esta interfaz, se deben identificar cinco secciones desde las cuales se irá trabajando en la programación.



La primera sección corresponde a las dos barras de menú en la parte superior, ubicadas de forma horizontal (la primera con menú compuesto de palabras, y la segunda compuesta de íconos). En estas barras, hay un botón crucial que se recomienda que sea tenido en cuenta constantemente al ir programando el experimento: el control de Guardar. El programador debe ir guardando su archivo con las modificaciones realizadas, previniendo que ante cualquier interrupción de su trabajo, el archivo del experimento pueda ser perdido parcial o completamente. La sección dos se puede observar en la parte izquierda de la interfaz, una columna llamada 'Toolbox' (en español, Caja de herramientas). Aquí se encontrarán todos los objetos posibles de incluir en el experimento, y a los cuales se les irá modificando sus propiedades para manipular la presentación estimular. Los objetos se irán describiendo a lo largo de la parte 2 de esta guía, así como también sus principales usos y propiedades.

La sección tres se observa al costado derecho de la Caja de herramientas, y se llama 'Structure' (en español, Estructura), y aquí se irán observando todos los cambios realizados en la programación experimento. Cada vez que un objeto se añada al experimento, se observará su inclusión y eliminación en la estructura. Todo objeto eliminado de la estructura se conservará debajo de la carpeta dentro de la estructura llamada 'Unreferenced Objects' (en español, objetos no referenciados). El objeto no estará ligado al experimento, pero se mantiene allí. Si se quiere eliminar definitivamente del experimento, se debe eliminar también de esta carpeta. Para eliminar un objeto, se debe hacer click derecho sobre el objeto y se selecciona la opción eliminar. Se recomienda al programador que monitoree la estructura del experimento cada vez que realiza un cambio en el experimento. El objeto que se encuentra por defecto en la estructura es el que se llama 'Procedure' (en español, procedimiento). Este objeto representa la línea temporal del experimento y muestra cómo se irán presentado los objetos (algunos de los cuales serán instrucciones del mapping y estímulos) temporalmente desde el inicio hasta el fin del

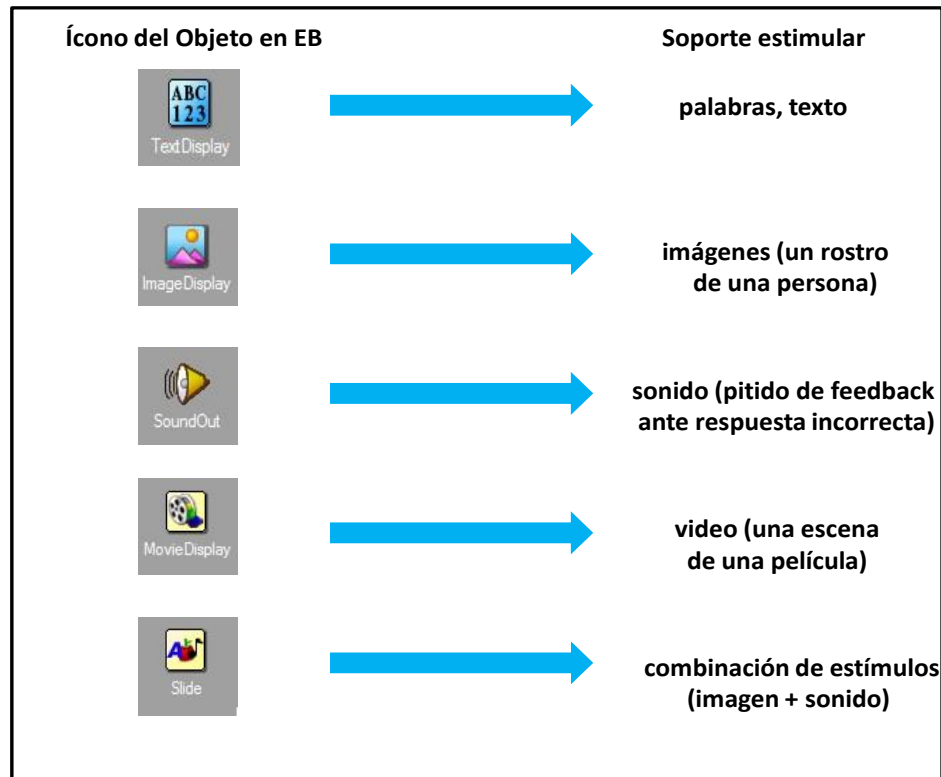
experimento. Concretamente, el experimento inicia desde el punto verde y finaliza en el punto rojo que se observan en la línea temporal.

La sección cuatro se encuentra al costado derecho de la Estructura, y se observa como un fondo gris simplemente. Cada vez que se 'abra' un ícono de un objeto, la interfaz de cada objeto se mostrará de forma completa en la sección cuatro. El procedimiento para abrir la interfaz de un objeto es hacer doble click sobre el ícono del objeto. La sección cinco se encuentra debajo de la pantalla, de forma horizontal, y se llama 'Output' (en español, Salida). En ese recuadro blanco se podrá ir observando todo lo generando mediante la POO en lenguaje BASIC. El usuario de EB puede modificar la interfaz gráfica a su modo, cerrando las secciones desde el ícono con una X en la parte superior derecha cuando aparezca, o modificando el tamaño de la sección desde los contornos de cada estructura. Desde el menú 'View' (en español, Vista) de la sección uno se puede modificar varios de estos aspectos.

A continuación, se mostrará cómo interactúan las secciones dos, tres y cuatro en la programación. Cada objeto de la caja de herramientas que se desee incluir en el experimento, deberá ser 'arrastrado y soltado' en el objeto procedimiento, en el lugar temporal que el programador desee incluirlo. Primero, se debe abrir el objeto procedimiento. Luego, se selecciona mediante un click izquierdo el objeto a arrastrar desde la Caja de herramientas, y manteniendo presionado el botón se moviliza el objeto hasta el procedimiento abierto en la sección cuatro, soltándolo (dejando de presionar el botón izquierdo) sobre la línea negra del procedimiento.

3. Los estímulos del experimento

Se describió en la parte 1 a los estímulos como las 'estrellas' del experimento: es lo que la persona percibirá y ante lo que se le pedirá que emita una respuesta. El siguiente cuadro resume cuáles son los objetos que pueden presentar estímulos en EB y qué tipo de estímulo presenta cada uno.

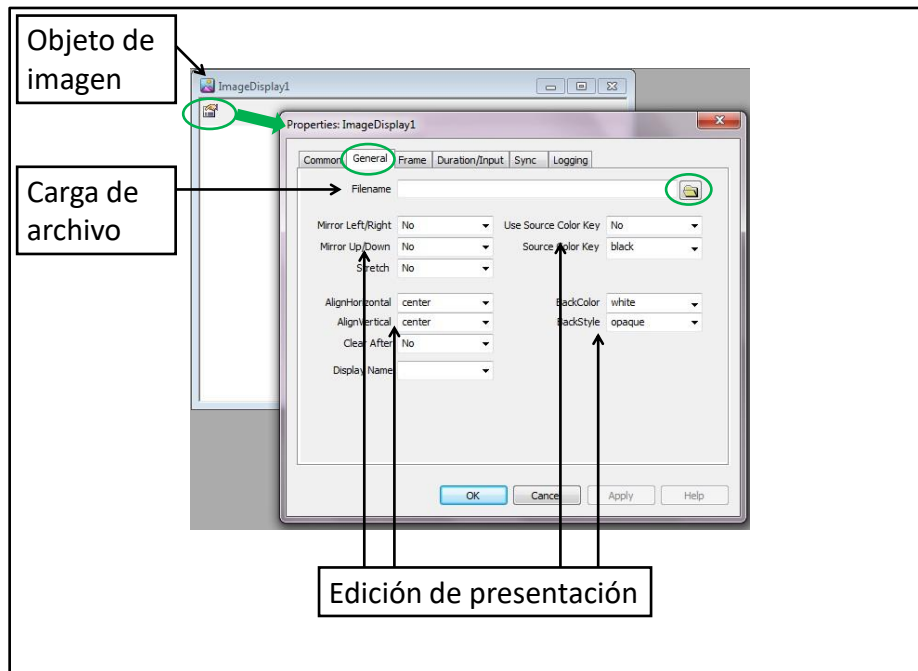


Para que un estímulo aparezca en el experimento, debe ser ‘arrastrado’ al procedimiento, esto es, a la línea temporal del experimento. La ‘edición’ de cada estímulo se realiza desde las propiedades de cada objeto. El botón para acceder a las propiedades se encuentra en la parte superior izquierda de cada objeto, y tiene la siguiente forma: . Desde la interfaz de propiedades se manipularán todos los aspectos de la presentación.

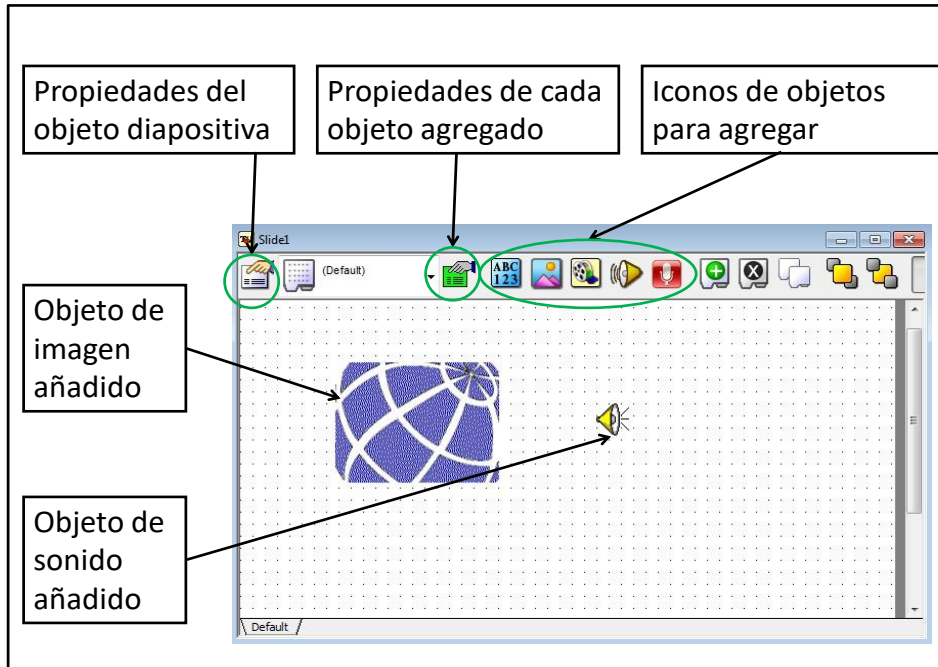
Existen propiedades comunes que se pueden editar en todos los objetos: el nombre, la duración, qué respuesta recolectan, de qué forma. Pero cada objeto también posee propiedades específicas del soporte estimular que presenta. Por ejemplo, en el objeto de texto se puede editar el color y fuente de las palabras escritas. En el objeto de sonido se puede configurar el volumen y la lateralidad de salida del sonido en los parlantes o auriculares conectados al ordenador. En el objeto de video, se puede seleccionar un fragmento específico del video que será presentado como estímulo.

Hay que tener en cuenta un aspecto importante en la programación en EB para presentar un estímulo. En el objeto de texto, las palabras pueden escribirse directamente en el objeto y serán presentadas en la pantalla. Pero el resto de los objetos para presentar estímulos poseen una variación fundamental al objeto de texto: necesitan de un archivo generado externamente a EB. Si se desea presentar imágenes, sonidos o videos, se debe generar y/o tener el archivo en el ordenador, y luego incluirlo en el objeto correspondiente. Para el correcto funcionamiento del experimento, es recomendable que el archivo del experimento programado y los archivos de

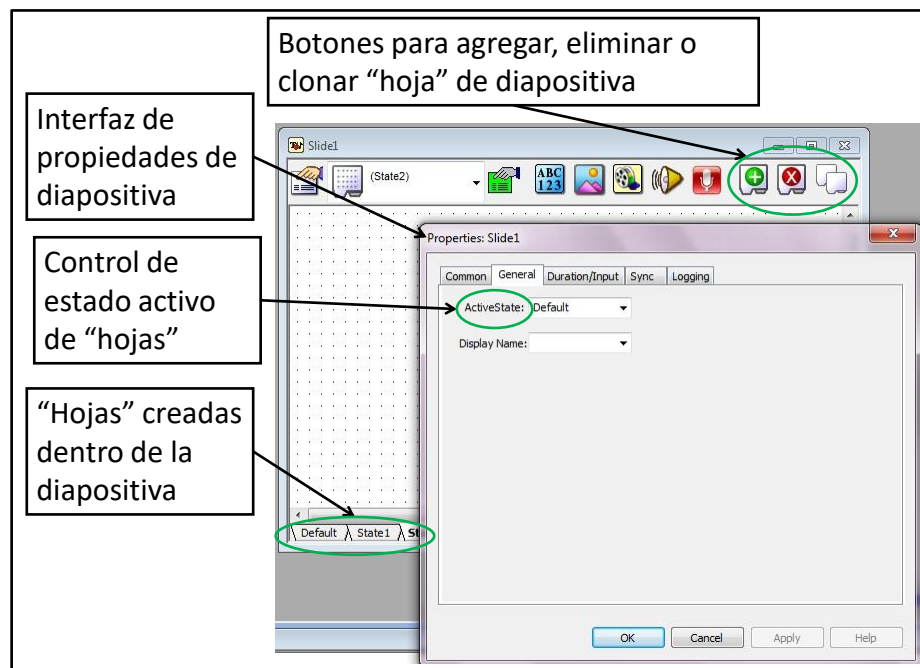
imágenes, sonidos y/o videos, sean guardados dentro de la misma carpeta en el ordenador. Esto facilitará la recuperación de archivos para su uso en el experimento. Para incluir el archivo de imagen, video o sonido en el objeto correspondiente, se debe abrir el menú General en las propiedades del objeto utilizado, y en el campo 'Filename' (nombre del archivo) escribir el directorio donde se encuentra el archivo en el ordenador, o directamente buscarlo presionando el ícono de la carpeta que se encuentra a la altura del campo 'Filename'. En la siguiente figura se observa un ejemplo de cómo ingresar a las propiedades de un objeto de imagen, dónde cargar un archivo externo y cómo comenzar a editar sus propiedades.



Hubo un objeto para presentar estímulos al que se lo mencionó con cierta consideración en la parte 1: el objeto de diapositiva ('Slide'). Cuando es necesario presentar varios soportes estímulares de manera combinada, o varios estímulos del mismo tipo, el objeto diapositiva permite manipular todo tipo de combinaciones. La interfaz del objeto diapositiva se muestra en la figura que sigue a continuación, y se puede observar que el objeto tiene un propio menú en su interfaz. El primer botón del menú corresponde a las propiedades del objeto, en donde se encuentra por ejemplo el menú de Duración y entrada de datos. Luego se puede observar el mismo botón de propiedades, pero de color verde: este botón corresponde a las propiedades de cada objeto que se vaya agregando a la diapositiva. Para agregar un objeto, se debe clicar sobre el objeto dentro del menú de la diapositiva (los íconos aparecen posteriormente al botón de propiedades de objetos), y luego clicar sobre el fondo de la interfaz de la diapositiva. Una vez agregado el objeto (por ejemplo, un objeto de imagen), si se cliquea sobre el objeto de imagen y luego se cliquea sobre el botón de propiedades 'verde', la interfaz de propiedades que se mostrará será solamente la específica para el objeto de imagen, y no para la diapositiva o el resto de los objetos agregados.



Al costado de los íconos de objetos para agregar, se observan tres botones referidos a la diapositiva: para agregar, eliminar o clonar (duplicar) la diapositiva. El objeto de diapositiva permite, además de integrar objetos, generar varias diapositivas dentro del mismo objeto. Las diapositivas que se vayan adicionando o duplicando, se observarán agregadas como 'hojas' en la parte inferior izquierda del objeto. El mismo cambio se observará si se elimina una de estas 'hojas'. El control sobre qué hoja de la diapositiva será utilizada se realiza desde las propiedades generales del objeto diapositiva (botón 'blanco' de propiedades), en el menú General, modificando el campo 'Activate Sate' (en español, estado activo). En la siguiente figura se puede observar la interfaz del menú General de las propiedades de la diapositiva y los botones para añadir 'hojas' a la diapositiva.



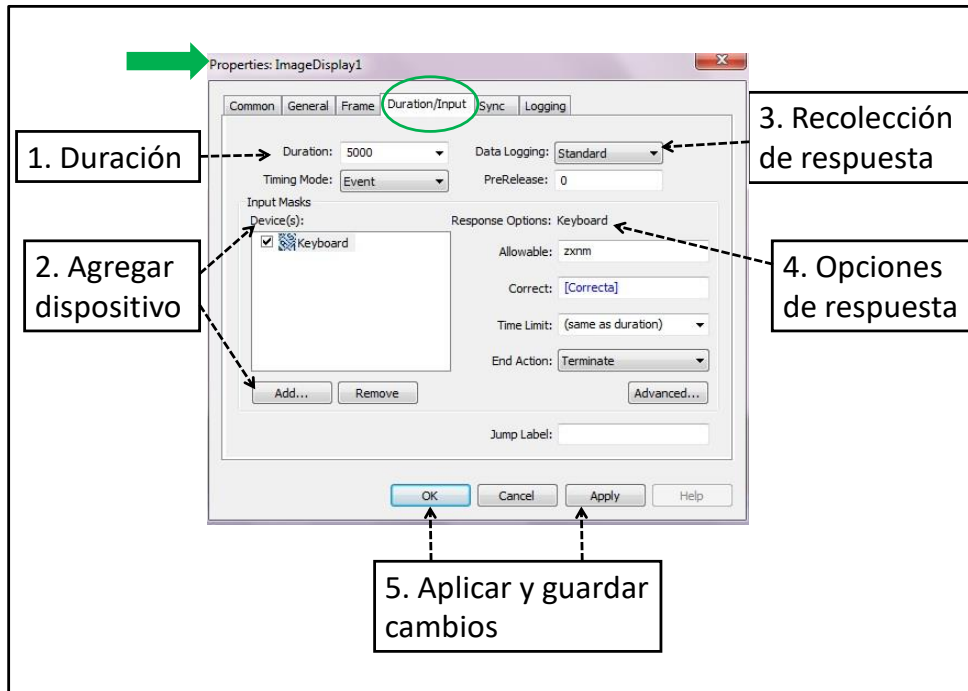
Repasemos hasta el momento qué aspectos del **checklist** para programar un experimento en psicología se han solucionado:

ITEM	CONTENIDO	ESTADO
5	¿Qué tipo de estímulos van a ser presentados? ¿Son imágenes, sonidos, diapositivas, palabras, videos? ¿Algún otro?	✓

4. La recolección de respuestas

En la parte 1 se presentó un aspecto crucial del diseño de experimentos en psicología: la recolección de datos conductuales. Se mencionaron dos valores 'claves' a ser recolectados: el tiempo de reacción (TR) y el *accuracy* (ACC). Ahora se explicará cómo desde EB se puede programar la recolección de estas respuestas.

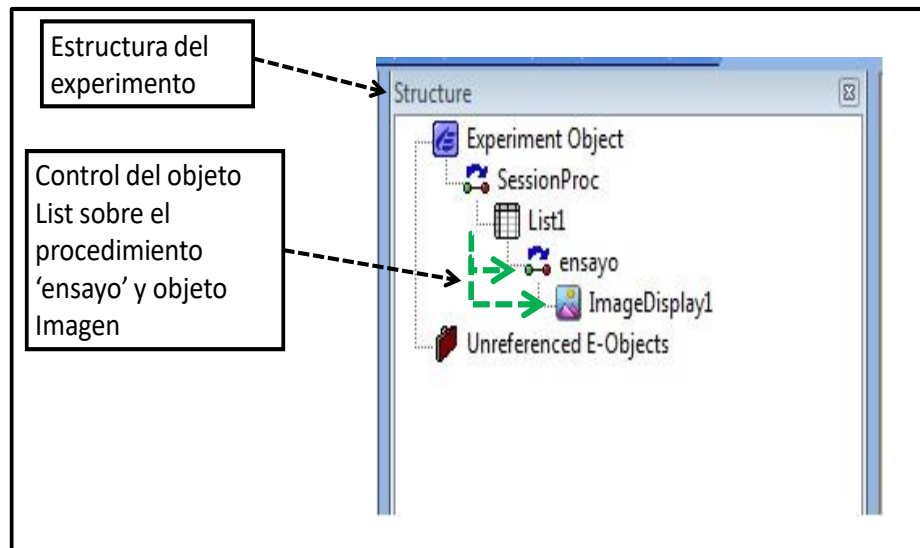
Se debe acceder al menú de Propiedades del objeto que presentará el estímulo y sobre el que se emitirá la respuesta. Una vez abierta la interfaz de propiedades, se selecciona la ventana 'Duration/Input' (duración y entrada de datos), que se observa en la siguiente imagen:



Una vez en esa ventana, el procedimiento para configurar la recolección de respuestas es el siguiente:

1. En el campo de duración se escribirá el valor en milisegundos que durará la presentación del objeto (por ejemplo, 5000 ms).
2. Para recolectar los datos de la respuesta, es necesario agregar un dispositivo de entrada (en dónde va a emitir la respuesta la persona). Se puede seleccionar desde el botón 'Add' el teclado o el mouse.
3. Seleccionando el formato de recolección 'Standard', se obtienen los datos de TR, ACC, respuesta emitida, entre otros, ante cada respuesta.
4. En opciones de respuesta, se introduce:
 - En el campo 'Allowable', las teclas que pueden presionarse. En este caso se ingreso las teclas z, x, n, m, sin comas, una al lado de la otra (zxnmm), teniendo en cuenta el experimento de presentación de rostros en marcos de colores descrito en la parte 1.
 - En el campo 'Correct' se introducen las teclas que serán admitidas como correctas. En este ejemplo, se escribió un atributo, entre corchetes, para vincular cada tecla con un tipo de estímulo y así establecer cuál es la respuesta correcta en cada estímulo. Esto será explicado a continuación.
5. Por último, se presionan los botones 'Apply' y 'Ok' para aplicar y guardar los cambios generados.

Hasta ahora se ha logrado obtener la recolección de TR y la respuesta emitida por la persona. Cada vez que se presione una tecla ante el estímulo, que durará 5000 ms, EB recolectará el TR entre el estímulo y la ejecución de la respuesta en milisegundos. Pero, ¿cómo se determina cuál es la respuesta correcta y cuáles las incorrectas para cada estímulo presentado? Para ello se utilizó una vinculación mediante atributos entre dos objetos. El objeto List controla los procedimientos y los objetos que presentan estímulos y recolectan respuestas. En la siguiente imagen se puede ver en la línea temporal del experimento que primero se encuentra el objeto List, para controlar el procedimiento de presentación de Imagen.



En el objeto List se añadirán los atributos que sean necesarios para caracterizar cada estímulo y la respuesta correcta esperada. El objeto List presenta una matriz para ser completada con el diseño pensado por el experimentador. Un atributo en el objeto List es una columna de la matriz, y se

añade pulsando el siguiente botón: . El atributo se nombra en la primera fila de cada columna. Se debe ingresar como atributo en el objeto List toda característica del estímulo que deba ser controlada en los objetos que presentan los estímulos. Por ejemplo, si el marco del rostro de la persona cambia de color, este cambio se puede controlar desde el objeto List: se ingresa un atributo para establecer el control. La matriz del objeto List está compuesta de columnas (atributos) y de 'niveles'. Un nivel es una fila de la matriz, y difiere de otro nivel por el contenido

que cada uno tiene. Se añade pulsando el siguiente botón: . Las diferentes combinaciones que se producen en cada nivel describen las 'condiciones' de la presentación estimular (a veces sólo un nivel describe una condición, en otros casos varios niveles describen cada condición).

Por ejemplo, un nivel puede ser completado con la siguiente condición: 'rostro mujer – edad joven – marco verde', y el siguiente nivel contiene una condición diferente: 'rostro varón – edad avanzada – marco morado'.

Para establecer una vinculación lógica entre el atributo ingresado en el objeto List y una propiedad de algún otro objeto del experimento, simplemente es necesario escribir en la propiedad del objeto el atributo entre corchetes (como en el ejemplo visto anteriormente con la palabra [\[Correcta\]](#)). Entonces, en el objeto List se especificará la siguientes ‘condiciones’ de presentación de estímulos, y la vinculación de estímulos y respuestas correctas:

Imagen	Género	Edad	Marco	Correcta
1.bmp	Mujer	joven	verde	x
2.bmp	Varón	avanzada	morado	m
1.bmp	Mujer	joven	morado	z
2.bmp	Varón	avanzada	verde	n

En la tabla, que será completada de esa misma forma en la matriz del objeto List, se observa que sólo una de las cuatro teclas permitidas aparece como respuesta correcta. Si se presiona alguna de las otras tres teclas permitidas como respuesta en un estímulo, la respuesta será clasificada como incorrecta. Tal como se describió el experimento en la parte 1, si el marco es morado, se responde con la tecla ‘z’ ante el rostro de edad joven y con la tecla ‘m’ ante el rostro de edad avanzada; si el marco es verde, se responde con la tecla ‘x’ al género hombre de la persona, y con la tecla ‘n’ al género mujer de la persona. Cualquier otra tecla de las permitidas presionada en cada estímulo, será una respuesta incorrecta.

¿Y cómo sería diseñada la matriz en el caso que la respuesta correcta sea la ‘omisión’ de respuesta? La parte 1 mencionó un ejemplo concreto de un experimento en el cual se debía responder sólo ante algunos eventos, pero en la mayoría no se debía emitir respuesta alguna: el experimento de Visual Tracking (presentación de círculos rojos consecutivos y no consecutivos). En este caso, la programación modificaría los siguientes aspectos:

- En el campo ‘Allowable’, sólo correspondería una tecla de respuesta posible: la barra espaciadora. El formato para escribir teclas especiales en EB es en mayúsculas y entre llaves, de la siguiente forma: {SPACE}.
- En el campo ‘Correct’, continuamos utilizando la vinculación lógica de atributos entre el objeto List y el objeto de Imagen, mediante la escritura del atributo [\[Correcta\]](#).
- La ‘clave’ nuevamente está en la matriz del objeto List. En los estímulos que la presentación de círculos no sea consecutiva, la respuesta correcta será la presión de la barra espaciadora, mientras que cuando la presentación si sea consecutiva, la respuesta correcta será la ‘omisión’ de respuesta: en este último caso, la celda del atributo Correcta debe dejarse en blanco. La tabla que se muestra a continuación (matriz para el objeto List) ilustra el ejemplo:

Imagen	Número de Círculo	Consecutivo	Correcta
1.bmp	uno	si	
2.bmp	dos	si	
3.bmp	tres	no	{SPACE}

Repasemos el **checklist** para programar un experimento en psicología nuevamente y observemos que aspectos se han solucionado y cuáles pueden haber quedado en suspenso:

ITEM	CONTENIDO	ESTADO
5	¿Qué tipo de estímulos van a ser presentados? ¿Son imágenes, sonidos, diapositivas, palabras, videos? ¿Algún otro?	✓
6	¿Se van a presentar diferentes condiciones en la presentación estimular? ¿Cuáles?	✓
7	¿Qué atributos componen los estímulos?	✓
8	¿Qué tipo de respuesta quiero recoger?	✓
20	¿Qué atributos explícitos conforman la estructura de los estímulos?	✓
21	¿Cuáles son los atributos implícitos que debería incluir para mejorar el análisis de datos?	...

Cómo puede observarse en el **checklist**, las últimas páginas han trabajado el diseño de la matriz del objeto List para ingresar los atributos de los estímulos y vincularlos lógicamente con otros objetos del experimento. Pero, ¿es necesario agregar atributos implícitos que no estén vinculados a otro objeto del experimento? Tal como lo indica el ítem 21 del **checklist**, agregar atributos implícitos a la matriz del objeto List ayuda a mejorar el análisis de datos posterior.

Para solucionar esta cuestión, utilizaremos el ejemplo de la tarea conductual para evaluar el uso del mecanismo de reappraisal (regulación emocional) mediante la presentación de palabras. La presentación estimular contiene cuatro palabras en total: una palabra clave, una palabra target y dos palabras distractores. La palabra clave y la palabra target están relacionadas semánticamente, de manera fuerte (mayor cercanía semántica) o débil (menor cercanía semántica). Además, cada palabra puede tener una valencia emocional positiva o negativa, generando congruencias o incongruencias afectivas entre las palabras presentadas. La matriz del objeto List con los atributos explícitos para ‘correr’ el experimento sería la siguiente:

Clave	Target	Distractor1	Distractor2
PAREJA	AMOR	VERDAD	MUSICA
ÁNGEL	DIABLO	TRAGEDIA	VENENO
AVENTURA	PELIGRO	MADRE	RESPETO
INUNDACIÓN	TRAGEDIA	IRRITACIÓN	LADRÓN

Pero es posible agregar a esta matriz atributos implícitos respecto a la distancia semántica y la congruencia afectiva entre los componentes de la tarea (cue, target y distractores), lo que dejaría la matriz completada de la siguiente forma:

Clave	Target	Distractor1	Distractor2	Dist.Sem. C-T	Condic. Afectiva	Congr.Afec Cue-Target	Congr. Afec. Cue- Dist
PAREJA	AMOR	VERDAD	MUSICA	1	111	1	1
ÁNGEL	DIABLO	TRAGEDIA	VENENO	1	122	2	2
AVENTURA	PELIGRO	MADRE	RESPE TO	2	121	2	1
INUNDACIÓN	TRAGEDIA	IRRITACIÓN	LADRÓN	2	222	1	1

Se percibirá que los atributos implícitos agregados fueron completados con números en cada nivel de la matriz: son ‘códigos’ o ‘claves’ que utiliza el experimentador para representar las relaciones entre las características de un estímulo. Si la distancia semántica entre el cue y el target es fuerte, se coloca un 1; si es débil, corresponde un 2. Si la palabra tiene una valencia emocional positiva corresponde un 1, y si es negativa un 2: de la combinación entre los tres componentes de la tarea surge la condición afectiva del estímulo. La congruencia afectiva entre dos palabras se codifica con un 1 y si hay incongruencia corresponde un 2.

Por cada nivel de la matriz, se obtendrá una respuesta: una tecla presionada, un TR y el ACC de cada respuesta. Lo que agregan los atributos implícitos son características del estímulo y de la condición a la que pertenece, que no ‘controlan’ la presentación del estímulo pero si mejoran el análisis de datos posterior.

Entonces, ahora el **checklist** para programar experimentos en psicología se encuentra en el siguiente estado:

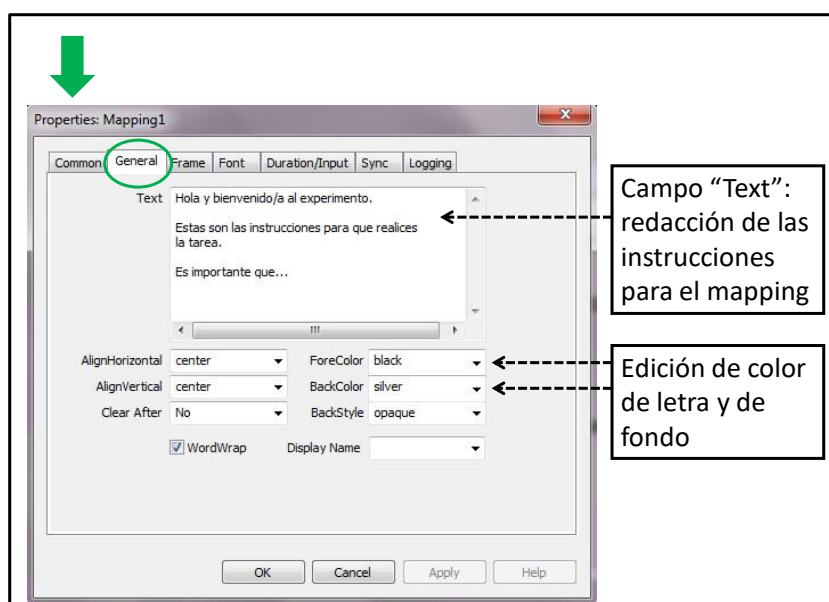
ITEM	CONTENIDO	ESTADO
5	¿Qué tipo de estímulos van a ser presentados? ¿Son imágenes, sonidos, diapositivas, palabras, videos? ¿Algún otro?	✓
6	¿Se van a presentar diferentes condiciones en la presentación estimular? ¿Cuáles?	✓
7	¿Qué atributos componen los estímulos?	✓
8	¿Qué tipo de respuesta quiero recoger?	✓
20	¿Qué atributos explícitos conforman la estructura de los estímulos?	✓
21	¿Cuáles son los atributos implícitos que debería incluir para mejorar el análisis de datos?	✓

5. Programar el mapping

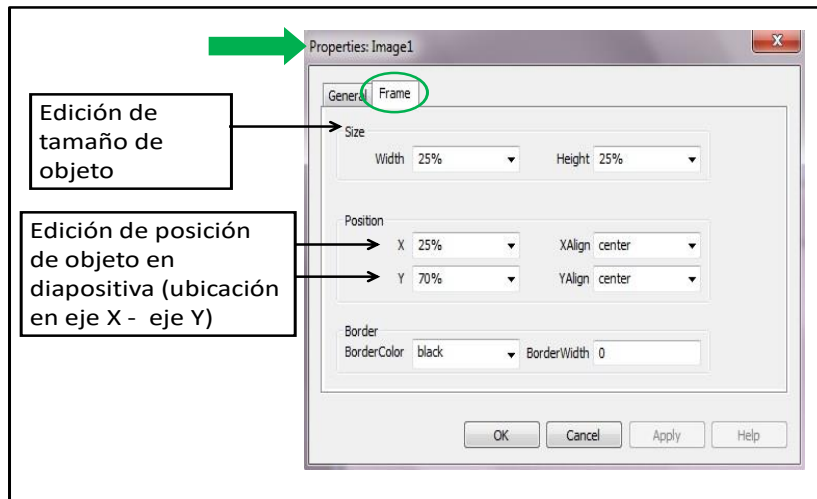
El apartado 3 y 4 de la parte 1 tienen como eje principal el ‘mapping’: son las instrucciones que recibe la persona para poder realizar luego el experimento de la mejor manera posible. En el diseño del experimento, los objetos que correspondan al mapping deberían aparecer en el primer lugar de la línea temporal del experimento. Será lo primero que reciba y visualice la persona.

El objeto que será elegido para incluir el mapping es el objeto de texto. Si el experimentador considera que a las instrucciones textuales es necesario añadirle ejemplos de los estímulos que aparecerán posteriormente, y estos estímulos son por ejemplo imágenes, el objeto de EB seleccionado para incluir el mapping será una diapositiva (combinación de texto e imágenes).

Sea un objeto de texto o diapositiva, los aspectos para programar el mapping en EB son sencillos. En el objeto se incluyen las instrucciones redactadas. La forma de redacción fue trabajada en profundidad en el apartado 4 de la parte 1. Para escribir las instrucciones, se ingresa a las propiedades del objeto de texto, ventana 'General', campo 'Text': allí se puede escribir lo que luego aparecerá en la pantalla. Como se mencionó anteriormente, se puede editar el formato de la presentación: se suele utilizar un color de letra negro (desde campo 'Fore color') y color de fondo gris (desde campo 'Back color') para presentar el mapping.



Si se combinan texto e imágenes en las instrucciones, será importante que queden 'alineados' en el objeto diapositiva: se edita la 'posición' de cada objeto en la diapositiva para que no se superpongan entre sí. Esto se configura desde las propiedades de cada objeto, en la ventana 'Frame' (en español, marco): se modifican los valores de ubicación de cada objeto en los eje X e Y, así como el tamaño de cada objeto.



Desde las propiedades del objeto, en la ventana 'Duración/entrada de datos', se colocan los siguientes valores (siguiendo la figura presentada en el apartado 2 previamente):

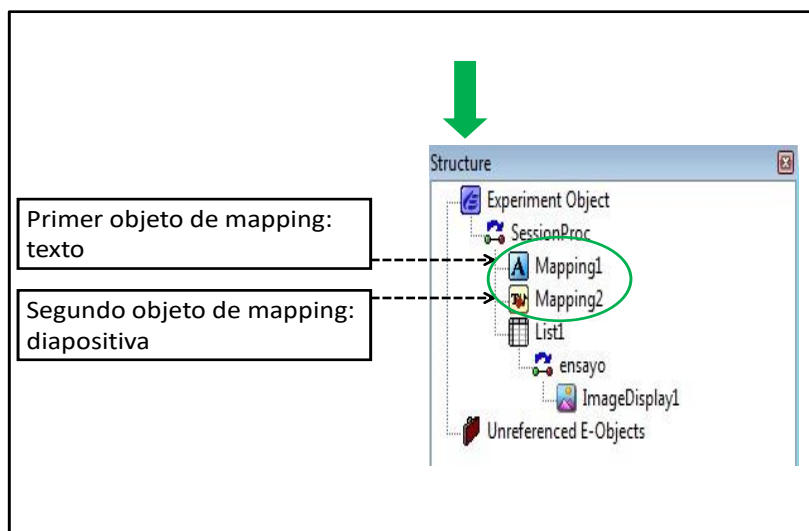
1. En el campo de duración, se puede optar por dos valores según lo explicado en la parte 1. Si se asigna un tiempo específico, se recomendó en la parte 1 una duración de 15000 ms. Si las instrucciones permanecerán en la pantalla hasta que la persona crea que ha comprendido la tarea, se ingresa entre paréntesis la palabra (infinite), que establece una duración infinita.
2. Si la duración del objeto fue programada como infinita, se añadirá un dispositivo de entrada para finalizar la aparición del objeto. Desde el botón 'Add' se seleccionará el teclado.
3. Se seleccionará el formato de recolección '(none)', ya que en este objeto no es necesario obtener los datos de TR, ACC y respuesta emitida.
4. En el caso que la duración del objeto sea infinita y se necesite pulsar una tecla para finalizar el objeto, en 'Opciones de respuesta', se introduce:

- En el campo 'Allowable', la tecla que deberá ser presionada para finalizar el objeto. Se puede elegir que la persona presione cualquier o una tecla específica. Si se opta por la primera opción, se escribe entre llaves la palabra {ANY}. En el caso que sea una tecla específica, se escribe por ejemplo la palabra {SPACE} para la barra espaciadora.
- En el campo 'Correct' no será necesario incluir ninguna tecla, ya que no existen respuestas correctas e incorrectas.

5. Por último, se presionan los botones 'Apply' y 'Ok' para aplicar y guardar los cambios generados.

Se habrá observado que es posible añadir una tecla que será pulsada para finalizar el objeto del mapping. A veces el mapping puede estar programado en varios objetos de texto o diapositivas, dependiendo de cuántos sean necesarios para explicar lo que se realizará durante la tarea. En ese caso, el pulsado de la tecla dará paso al siguiente objeto del mapping. Si es necesario presionar

una tecla en el objeto del mapping, entonces esa aclaración debe aparecer debajo de la redacción de las instrucciones: simplemente colocando al final del texto de cada objeto ‘Presiona la barra espaciadora para continuar’, la persona sabrá cómo continuar con las instrucciones siguientes o comenzar el experimento. La estructura del experimento se ha modificado con la inclusión del mapping, y se puede visualizar en la siguiente figura:



Repasemos el estado del **checklist** para ver qué aspectos nuevos se han trabajado para la programación del experimento:

ITEM	CONTENIDO	ESTADO
5	¿Qué tipo de estímulos van a ser presentados? ¿Son imágenes, sonidos, diapositivas, palabras, videos? ¿Algún otro?	✓
6	¿Se van a presentar diferentes condiciones en la presentación estimular? ¿Cuáles?	✓
7	¿Qué atributos componen los estímulos?	✓
8	¿Qué tipo de respuesta quiero recoger?	✓
9	¿Cómo haré para que las instrucciones sean comprendidas con claridad?	✓
10	¿El mapping será solo lectura de instrucciones o tendrá mi propia ayuda?	✓
20	¿Qué atributos explícitos conforman la estructura de los estímulos?	✓
21	¿Cuáles son los atributos implícitos que debería incluir para mejorar el análisis de datos?	✓

6. El bloque de práctica y el objeto ‘Feedback’

En la parte 1 de esta guía se explicó que cuando una persona hace un experimento puede recibir primero un entrenamiento y luego realizar los ensayos experimentales. El entrenamiento recibido está controlado por lo que se denomina un 'bloque de práctica'. En las siguientes páginas se explicará cómo programar este aspecto del experimento.

El bloque de práctica debe incluirse en la estructura del experimento (línea temporal) de forma posterior al mapping. Si corresponde un bloque de práctica, esto deberá ser aclarado de forma textual en algún objeto del mapping ('A continuación, se le presentarán una serie de ensayos de práctica'). Las diferencias principales entre un bloque de práctica y el bloque de ensayos experimentales son:

- Que en el bloque de práctica, los datos de las respuestas (TR, ACC, respuesta emitida) no son incluidos en el posterior análisis de datos.
- La cantidad de eventos del bloque de práctica es sustancialmente menor a la cantidad de eventos en el bloque experimental. El número de eventos de práctica estará definido por el experimentador, teniendo en cuenta el modelo teórico utilizado y el objetivo del bloque de práctica.

Para programar el bloque de práctica es necesario:

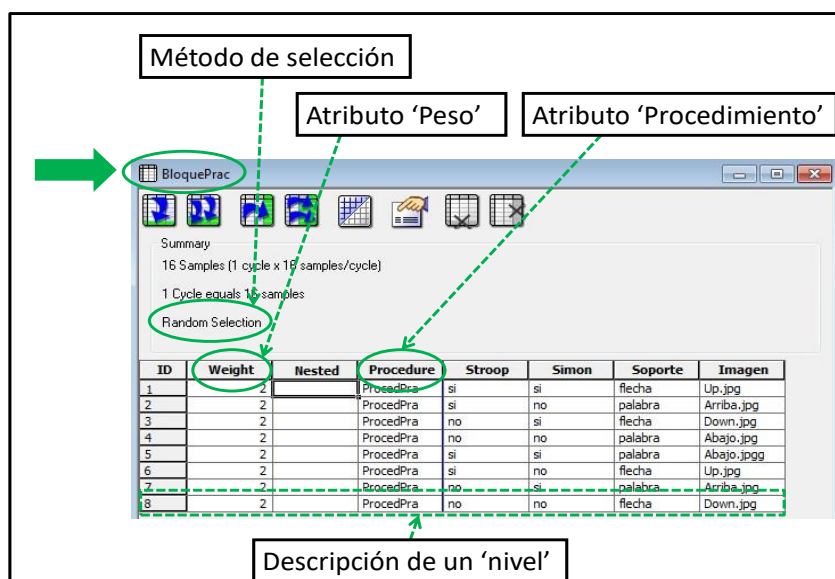
- Un objeto List que controle la presentación de eventos.
- Un procedimiento con los objetos que presentarán los estímulos de cada evento.

En el objeto List se configurarán todos los atributos de los eventos a ser presentados. En los niveles se incluirá cada ensayo (por ejemplo, cómo será la presentación de conflictos perceptuales Stroop y Simon con diferentes soportes estímulares). La inclusión de atributos implícitos no es necesaria en el bloque de práctica, pero puede depender de los objetivos que se haya propuesto el experimentador.

Para configurar el control de la presentación de los estímulos, se debe realizar lo siguiente:

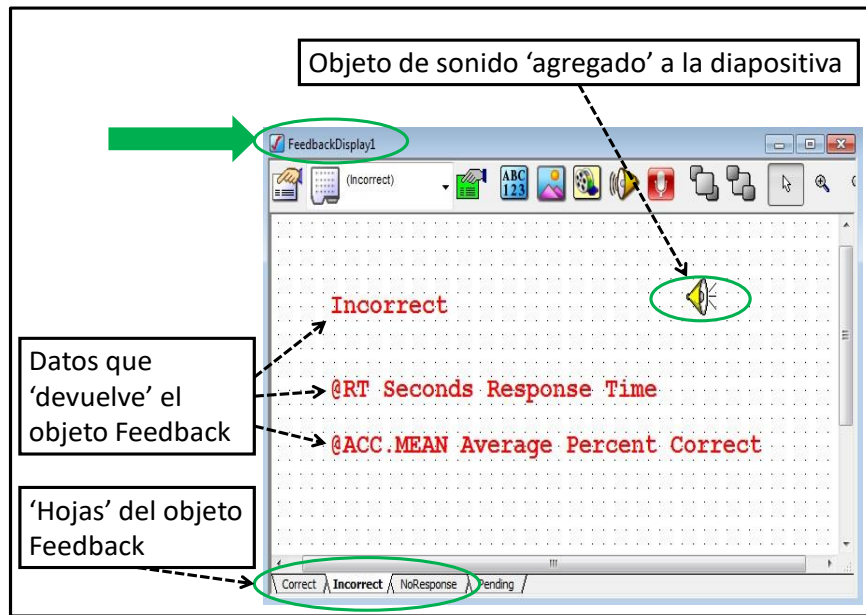
1. Desde el atributo 'Weight' (en español, peso), que aparece en la primer columna de la matriz, se configura la cantidad de veces que se presentará cada nivel. El número que se incluya en la celda será la cantidad de veces que ese ensayo será presentado.
2. En el atributo 'Procedure' se debe escribir el nombre del procedimiento que controlará el objeto List, en este caso, el procedimiento que incluye los objetos que presentarán los estímulos del bloque de práctica.
3. Una vez definida la combinación de atributos y niveles del bloque de práctica, se debe configurar un 'método de selección' de cada nivel. Esto se configura desde las propiedades del objeto List, en la ventana 'Selection', se elige el método 'Random' (en español, aleatorizado).

Los métodos de selección serán profundizados en un apartado posterior de la parte 2. La selección aleatorizada controla que los eventos de cada nivel se presenten aleatoriamente. En la siguiente figura se puede observar cómo quedaría configurada la matriz del objeto List que controla el bloque de práctica. Cada nivel se presentará 2 veces. El orden de las 16 presentaciones estará completamente aleatorizado.



Para finalizar la programación del bloque de práctica, se incluirá el objeto de 'Feedback' (en español, retroalimentación). En la parte 1 de la guía se mencionó la importancia de este objeto, y los motivos de su inclusión en el bloque de práctica y su eliminación en los ensayos experimentales. Ahora se aprenderán a programar estos aspectos.

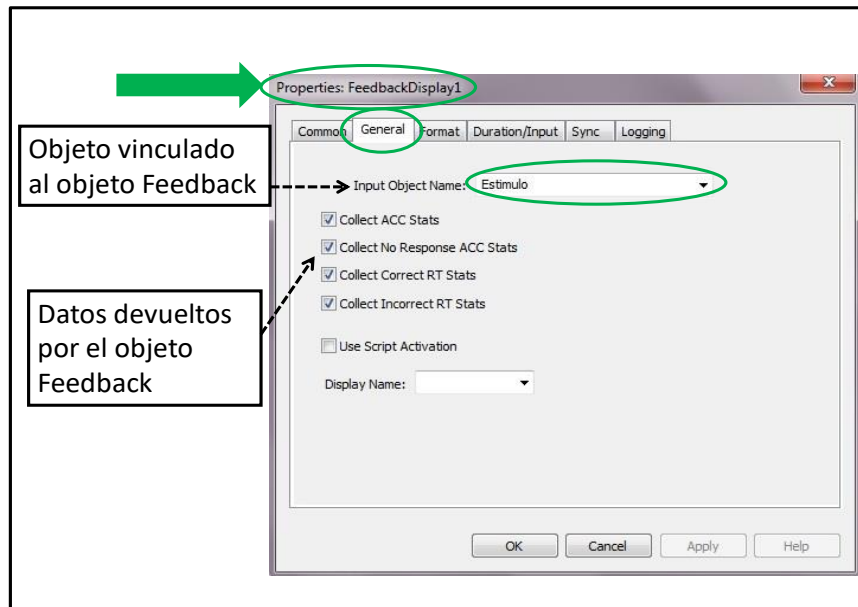
El objeto de Feedback es básicamente un objeto de diapositiva que contiene 'hojas' programadas con diferentes 'devoluciones' sobre la respuesta de la persona: respuesta correcta, respuesta incorrecta y respuesta no detectada. Ante cada respuesta, 'devuelve' los TR y ACC registrados.



Lo primero que se debe hacer para incluirlo es arrastrar el objeto al procedimiento de práctica en el lugar posterior del objeto de presentación de estímulos. De esta forma, luego de la aparición de cada estímulo y la emisión de la respuesta de la persona, aparecerá el feedback de su rendimiento.

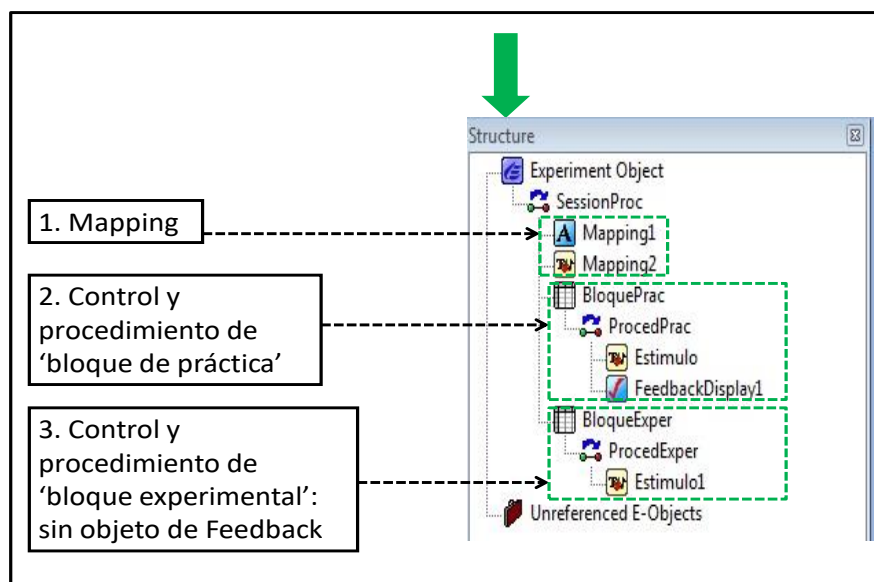
Para configurar el objeto de Feedback, se debe hacer lo siguiente:

1. En las propiedades del objeto, ventana 'General', campo 'Input Object Name' (en español, nombre de objeto de entrada), se debe ingresar el nombre del objeto que presenta los estímulos y recolecta la respuesta de la persona. En este campo se programa la 'vinculación' entre la emisión de la respuesta y el feedback devuelto.
2. La ventana de propiedades 'Duración/entrada de datos' está configurada por defecto: el objeto se presentará por 1500 ms y no recolectará ninguna respuesta ni será necesario presionar alguna tecla para su desaparición.
3. Es posible añadir un 'pitido' ante la respuesta incorrecta, de modo que acompañe al feedback textual programado en el objeto. Para ello, se debe ingresar en la 'hoja' de 'respuesta incorrecta' de la diapositiva del objeto de Feedback, y añadir un objeto de sonido (ver figura anterior). En las propiedades específicas del objeto de sonido, se debe añadir el archivo de sonido que será reproducido y configurar su volumen y duración. Se recomienda utilizar una duración breve, entre 50 y 100 ms, y un volumen moderado, que no 'asuste' a la persona por haber respondido de forma incorrecta. Los métodos para realizar estos procedimientos fueron explicados previamente, cuando se describió el objeto diapositiva.



La estructura del experimento se ha vuelto a modificar, y debe visualizarse como lo muestra la siguiente figura. Se puede observar que el objeto de Feedback sólo aparece en el procedimiento del bloque de práctica: es porque en los ensayos experimentales, ya no será necesario darle feedback a las personas. Simplemente el objeto no es incluido en el procedimiento.

Se podrá observar que los nombres de los objetos se han cambiado y que denominan la 'función' que cumple cada objeto en el experimento. Esto es recomendable cuando la estructura del experimento se vuelve más compleja, para poder identificar cada objeto desde el nombre. El nombre se puede modificar desde las propiedades del objeto, en la ventana 'Common', campo 'Name', o simplemente seleccionando cada objeto en la estructura y presionando la tecla 'F2'.



Revisemos una vez más el **checklist** para programar un experimento en psicología, para ver qué aspectos se han solucionado hasta el momento y cuáles han sido presentados y deben ser profundizados:

ITEM	CONTENIDO	ESTADO
5	¿Qué tipo de estímulos van a ser presentados? ¿Son imágenes, sonidos, diapositivas, palabras, videos? ¿Algún otro?	✓
6	¿Se van a presentar diferentes condiciones en la presentación estimular? ¿Cuáles?	✓
7	¿Qué atributos componen los estímulos?	✓
8	¿Qué tipo de respuesta quiero recoger?	✓
9	¿Cómo haré para que las instrucciones sean comprendidas con claridad?	✓
10	¿El mapping será solo lectura de instrucciones o tendrá mi propia ayuda?	✓
11	¿Necesito presentar un bloque de ensayos previos para asegurar que se comprendió la consigna?	✓
12	La presentación: ¿implica diferentes bloques?	...
13	¿Debería darle feedback a las personas? Si debería hacerlo, ¿será durante todo el proceso?	✓
14	¿Cuántos eventos quisiera presentar?	...
17	¿La presentación de los eventos será aleatoria?	...
20	¿Qué atributos explícitos conforman la estructura de los estímulos?	✓
21	¿Cuáles son los atributos implícitos que debería incluir para mejorar el análisis de datos?	✓

7. El control de la presentación de eventos

Los ítems del **checklist** que quedaron ‘en suspenso’ en el apartado anterior y los que se resolverán a continuación, tienen sus respuestas en un objeto se mencionó con especial atención en la parte 1: el objeto List. Hasta el momento se ha explicado cómo está compuesta la matriz del objeto List en atributos y niveles, y algunas de sus propiedades. En las siguientes páginas se profundizará en otros aspectos del uso de este objeto.

Empecemos con una propiedad del objeto que fue mencionada en el apartado anterior: el ‘método de selección’. En la ventana ‘Selection’, campo ‘Order’, se elige de qué manera el objeto ordenará la presentación de eventos. Se explicarán tres de estos métodos, los cuales se refieren a algunos de los diseños de programación explicados en la parte 1. Estos son:

1. Aleatorización ('random'): los eventos se presentan en un orden aleatorio desde el primero hasta el último, sin ningún criterio específico de selección.

2. Secuencial ('sequential'): la presentación de eventos respetará el orden en que se encuentran los niveles en la matriz del objeto List: se comienza por el primer nivel, luego el segundo, y así sucesivamente hasta que llegar al último nivel.

3. Contrabalanceo ('counterbalance'): implica alternar un orden seleccionado según un criterio específico, por ejemplo según la persona que realiza el experimento. Este método suele ser utilizado para controlar la asignación de las teclas de respuesta que utilizará cada persona. El experimento de presentación de conflictos perceptuales Stroop y Simon con diferentes soportes estímulares será útil para explicar un contrabalanceo por persona. Las teclas de respuesta 'permitidas' para responder en la tarea son la 'c' y la 'm', ante los estímulos Arriba y Abajo (tanto flecha como palabra)

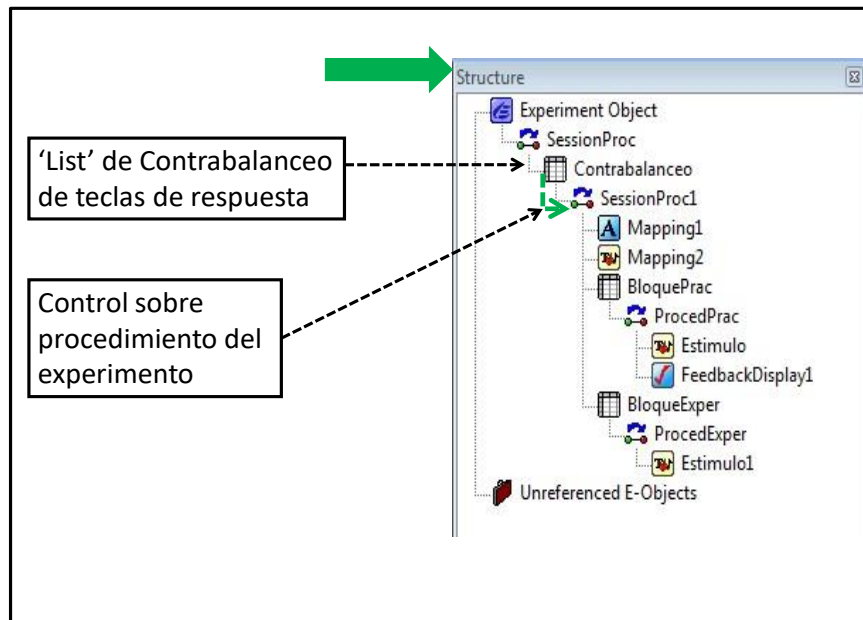
- La primer persona que realice la tarea utilizará el orden 1 de teclas: presionará la tecla 'c' cuando aparezca una palabra o flecha que indique 'Arriba', y la tecla 'm' cuando aparezca una palabra o flecha que indique 'Abajo'.

- La segunda persona, utilizará el orden 2 de teclas: tecla 'm' cuando aparezca una palabra o flecha que indique 'Arriba', y tecla 'c' cuando aparezca una palabra o flecha que indique 'Abajo'.

- La tercera persona utilizará nuevamente el orden 1, la cuarta persona el orden 2, y así sucesivamente.

El método de Aleatorización y Secuencial se pueden seleccionar simplemente desde las propiedades del objeto List mencionadas anteriormente. Para utilizar un método de contrabalanceo que controle las teclas de respuesta, como el explicado anteriormente, se utilizará una 'vinculación lógica' entre atributos. El procedimiento para programar esto en EB es el siguiente:

1. Se debe 'arrastrar' un objeto List al inicio de la línea temporal del experimento que controle el primer procedimiento. En la siguiente figura se observa cómo quedará la estructura del experimento.

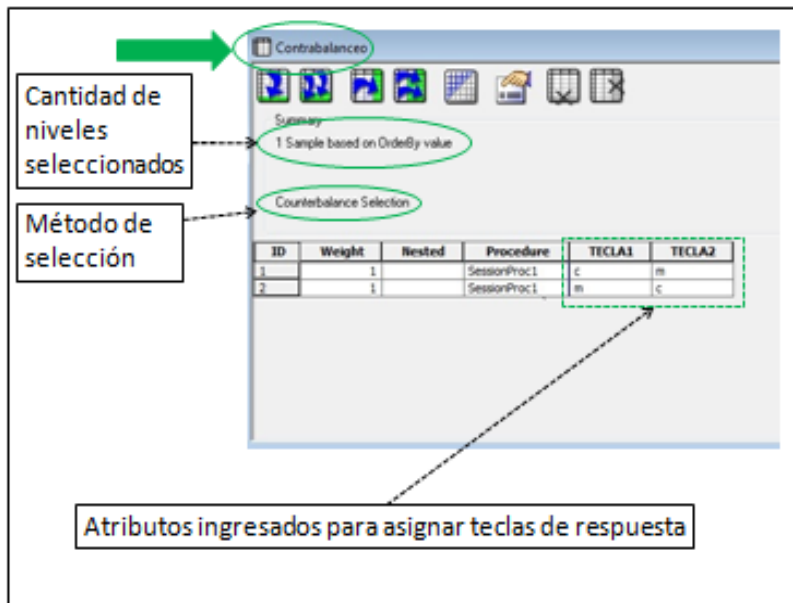


- La matriz de este objeto sólo estará compuesta por atributos que incluirán las teclas de respuesta. En el experimento de conflictos perceptuales Stroop y Simon, sólo serán dos atributos, que se llamarán 'TECLA 1' y 'TECLA2', sin comillas. Los niveles, serán las posibilidades de orden alternativos de las teclas. En este caso, sólo hay dos combinaciones posibles. En el nivel 1, será para 'TECLA 1' la letra 'c' y para 'TECLA2' la letra 'm'. En el nivel 2, se ingresará el orden inverso ('m' y luego 'c'). La matriz estará configurada sólo por dos atributos y dos niveles.

- En la ventana de propiedades 'Selection', campo 'Order', se debe seleccionar el método 'Counterbalance'. En el campo 'Order By' (en español, seleccionar por), elegir la opción 'Subject' (en español, sujeto).

-En la ventana de propiedades 'Reset/Exit' (en español, comenzar de nuevo y salir), se puede observar que EB configuró por defecto los campos, de modo que sólo se seleccione un nivel una sola vez, y luego finalice el experimento. Esto significa que será aplicado un orden de respuestas a todo el experimento y luego este finalizará; la siguiente persona utilizará el otro orden de respuestas establecido en el siguiente nivel para todo el experimento, y así sucesivamente.

La ventana principal del objeto List quedará configurada como muestra la siguiente figura.



2. Ahora se debe vincular los atributos del objeto List de contrabalanceo con las instrucciones del mapping. Las instrucciones deben mencionar las teclas de respuesta que se corresponden con cada estímulo presentado. Pero las teclas variarán según la persona realice la tarea ('contrabalanceo por sujeto'). Entonces, las instrucciones deberían quedar de la siguiente forma:

- 'Cuando se presente una flecha o palabra que indique el sentido ARRIBA, deberá presionar la [TECLA1], y cuando se presente una flecha o palabra que indique el sentido ABAJO, deberá presionar la [TECLA2]'

Se observará que 'TECLA1' y 'TECLA2' fueron escritos entre corchetes para ser referenciados como atributos. Esas palabras serán reemplazadas con el contenido de las celdas del nivel 1 de esos atributo para la primer persona que realice la tarea, y las celdas del nivel 2 de esos atributos para la segunda persona que realice la tarea, y así sucesivamente: concretamente, con las letras 'c' y 'm' según corresponda al contrabalanceo.

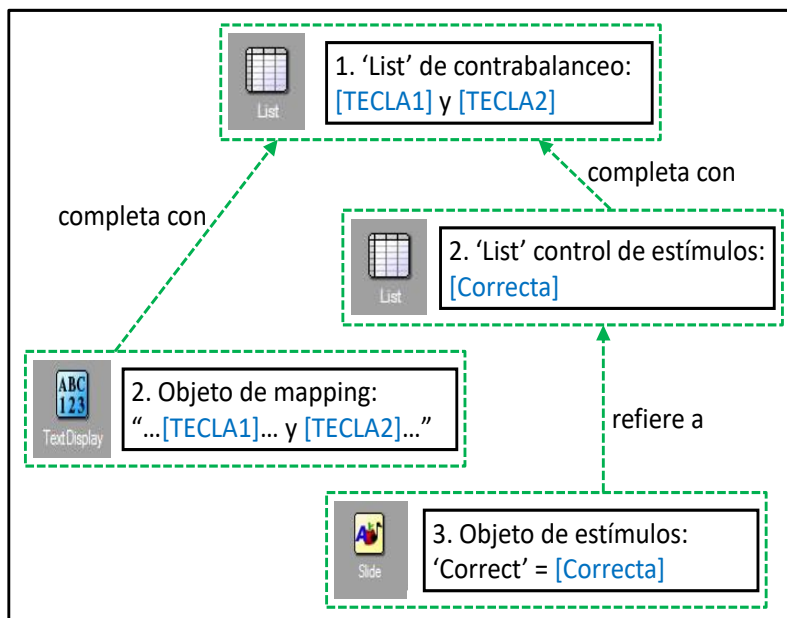
3. El último paso para programar el contrabalanceo de teclas por sujeto es 'vincular' el objeto List de contrabalanceo con los objetos List que controlan la presentación de estímulos.

- Ya se había explicado cómo vincular el campo 'Correct' de las propiedades de 'Duración/entrada de datos' de los objetos que presentan estímulos con el objeto List que los controla: completando el campo con un atributo [Correcta] y ese atributo del objeto List con las teclas de respuesta.

- Lo que debe realizarse ahora es reemplazar las celdas de los niveles del atributo 'Correcta' del objeto List que controla la presentación de objetos, con los atributos [TECLA1] y [TECLA2], según corresponda a cada evento. La siguiente figura ilustra a modo de ejemplo como debería ser completada la matriz del objeto List que controla la presentación estímulos (se observa principalmente la forma de completar los niveles del atributo 'Correcta'):

Nivel	Stroop	Simon	Soporte	Imagen	Correcta
1	si	si	flecha	Up.jpg	[TECLA1]
2	si	no	palabra	Arriba.jpg	[TECLA1]
3	no	si	flecha	Down.jpg	[TECLA2]
4	no	no	palabra	Abajo.jpg	[TECLA2]
5	si	si	palabra	Arriba.jpg	[TECLA1]
6	si	no	flecha	Up.jpg	[TECLA1]
7	no	si	palabra	Abajo.jpg	[TECLA1]
8	no	no	flecha	Down.jpg	[TECLA2]

La ‘vinculación lógica’ entre atributos utilizada para programar el contrabalanceo de teclas de respuesta por sujeto produjo una relación entre cuatro objetos: List de contrabalanceo, objeto de mapping, List de presentación de estímulos, objeto de presentación de estímulo. La siguiente figura ilustra la vinculación realizada.



Anteriormente se mencionó la propiedad ‘Reset/Exit’ del objeto List, la cual era configurada por defecto al seleccionar el método de contrabalanceo. En los métodos de selección secuencial o aleatorizado es necesario configurar los campos de esta propiedad. Desde esta ventana se determinará cuántos bloques y cuantos eventos por bloque serán presentados durante el experimento.

Una vez en la ventana ‘Reset/Exit’, se deberán configurar los siguientes campos:

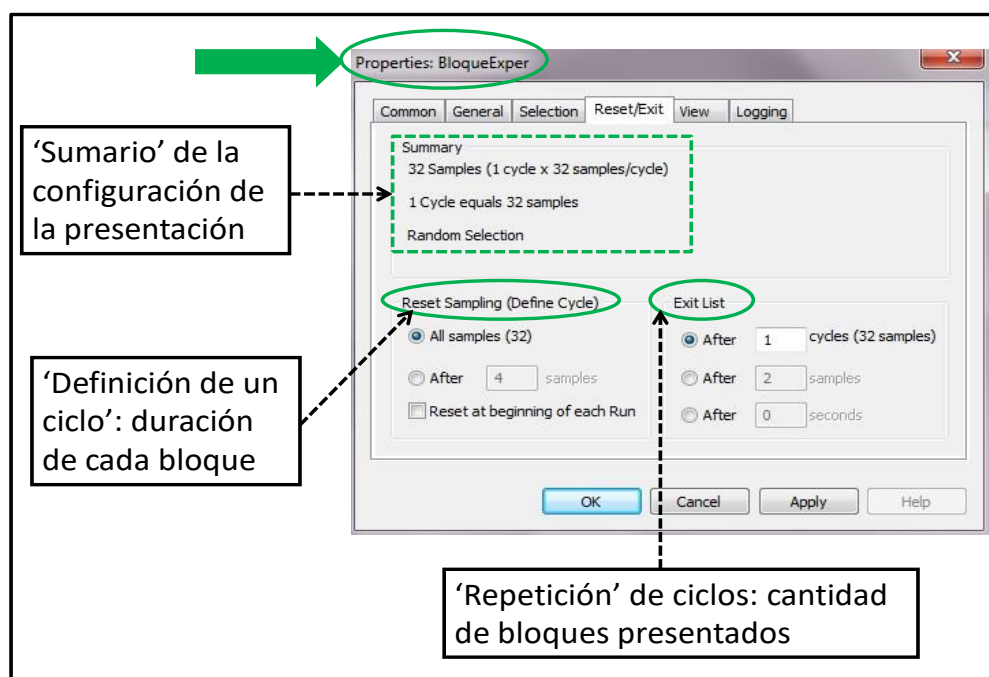
1. En el campo ‘Reset Sampling (Define Cycle)’ (en español, definir el ‘ciclo’) se editará la duración de cada bloque. Un ‘sample’ (en español, muestra) refiere a ‘un peso’ de cada nivel de la matriz

del objeto List (si el nivel tiene un 'Weight' = 5, esto equivale a 5 'samples'). Un 'ciclo' refiere a la cantidad de veces que se presentará cada evento.

- Para el experimento de conflictos perceptuales Stroop y Simpon, se seleccionará la opción 'All Samples'. Cada 'ciclo' finalizará cuando se presenten todos los eventos definidos en la matriz (todos los niveles por su 'peso' correspondiente). La matriz final para este experimento estará compuesta de 16 niveles presentados 2 veces cada uno, lo que equivale a 32 eventos por bloque.

2. En el campo 'Exit List', se configurará la cantidad de veces que se repetirá el 'ciclo' definido antes de finalizar el experimento. Esto equivale a programar la cantidad de bloques de la presentación.

- Se ingresará un valor de '1' (uno) en el espacio en blanco de 'After (...) cycles', y se observará que el número de cantidad de 'samples' indicado al lado de este campo seguirá siendo 32.



En el próximo apartado se mostrará cómo programar la aparición de más de un bloque experimental, añadiendo objetos de descanso para la persona que funcionen a modo de 'intervalo' entre los bloques experimentales.

Es momento de revisar el estado del **checklist** para programar experimentos en psicología y ver qué aspectos nuevos se han solucionado:

ITEM	CONTENIDO	ESTADO
5	¿Qué tipo de estímulos van a ser presentados? ¿Son imágenes, sonidos, diapositivas, palabras, videos? ¿Algún otro?	✓

6	¿Se van a presentar diferentes condiciones en la presentación estimular? ¿Cuáles?	✓
7	¿Qué atributos componen los estímulos?	✓
8	¿Qué tipo de respuesta quiero recoger?	✓
9	¿Cómo haré para que las instrucciones sean comprendidas con claridad?	✓
10	¿El mapping será solo lectura de instrucciones o tendrá mi propia ayuda?	✓
11	¿Necesito presentar un bloque de ensayos previos para asegurar que se comprendió la consigna?	✓
12	La presentación: ¿implica diferentes bloques?	✓
13	¿Debería darles feedback a las personas? Si debería hacerlo, ¿será durante todo el proceso?	✓
14	¿Cuántos eventos quisiera presentar?	✓
17	¿La presentación de los eventos será aleatoria?	✓
18	¿Es necesario presentar un tipo de contrabalanceo dentro del experimento?	✓
20	¿Qué atributos explícitos conforman la estructura de los estímulos?	✓
21	¿Cuáles son los atributos implícitos que debería incluir para mejorar el análisis de datos?	✓
22	¿Qué tipo de diseño de programación es el más adecuado?	✓

8. Programación ‘completa’ de un experimento de atención

En los apartados anteriores de la parte 2 de la guía se han explicado aspectos operativos de la programación de un experimento en EB. Se resolvieron los ‘desafíos’ propuestos en la parte 1 y los ítems más ‘operativos’ del **checklist** para programar un experimento en psicología. En este apartado, se tomará en cuenta el **checklist** completo y se responderá a cada ítem con la programación del experimento de atención de ‘presentación de conflictos perceptuales Stroop y Simon con diferentes soportes estímulares’.

- ✓ 1- Tener en claro el concepto que está ‘detrás’ del experimento.

Los Paradigmas de Asignación de Estímulos con Interferencias (PAEI) son utilizados comúnmente para el estudio de las redes atencionales y las estrategias cognitivas desarrolladas para la resolución de conflictos perceptuales. En este experimento se presentarán dos PAEI de forma singular y simultánea: el conflicto Stroop espacial y el conflicto Simon.

- ✓ 2- Repasar los modelos teóricos existentes sobre el concepto que se activará mediante el experimento.

Se tomará como referencia el modelo de Redes Atencionales de Posner (Posner & Petersen, 1989). Los resultados obtenidos tendrán implicaciones para el estudio de la Red de Control Ejecutivo Atencional (CEA).

- ✓ 3- ¿Qué deseo activar del sistema cognitivo de la persona?

La red CEA deberá implementar estrategias cognitivas de resolución de conflictos perceptuales para resolver los ensayos que presenten incongruencias Stroop espacial y Simon. El modelo de Monitoreo de Conflictos (Botvinick, Braver, Barch, Carter & Cohen, 2001; Botvinick, Cohen & Carter, 2004; Yeung, Botvinick & Cohen, 2004) propone que existen regiones corticales (cingulado anterior) que evalúan la potencialidad que tienen los estímulos para provocar conflictos perceptuales. Una vez detectado el conflicto, se envían señales al nodo de la corteza dorsolateral prefrontal, que elabora estrategias para disminuir la interferencia del conflicto.

- ✓ 4- ¿Qué antecedentes conozco que provoquen una activación parecida?

Existen múltiples antecedentes sobre la utilización de los PAEI a nivel conductual, imaginería cerebral, electrofisiológico y clínico (Hommel, 2010). La novedad del experimento consiste en utilizar diferentes soportes estímulares en la presentación de los PAEI (soporte iconográfico y verbal) para analizar si la red CEA implementa estrategias de resolución específicas del tipo de estímulo que presenta el conflicto.

- ✓ 5- ¿Qué tipo de estímulos van a ser presentados? ¿Son imágenes, sonidos, diapositivas, palabras, videos? ¿Algún otro?

Los estímulos estarán programados en un objeto de diapositiva y se presentará por cada evento una imagen de una flecha o una imagen de una palabra, que pueden indicar el sentido 'ARRIBA' o 'ABAJO'. Las palabras y flechas serán generadas en forma externa a EB en un formato de imagen (ver ejemplo de los estímulos en el experimento presentado en el apartado 7 de la parte 1). Las imágenes aparecerán en cada evento en una de cuatro posibles ubicaciones en la pantalla (esquinas).

- ✓ 6- ¿Se van a presentar diferentes condiciones en la presentación estimular? ¿Cuáles?

Las condiciones de la presentación estimular son tres, con dos variantes cada una: Stroop, Simon, soporte estimular (2 x 2 x 2). La siguiente tabla muestra las diferentes condiciones presentadas en el experimento:

Stroop	Simon	Soporte estimular
Congruente	Congruente	Iconográfico
Congruente	Congruente	Verbal

Congruente	Incongruente	Iconográfico
Congruente	Incongruente	Verbal
Incongruente	Congruente	Iconográfico
Incongruente	Congruente	Verbal
Incongruente	Incongruente	Iconográfico
Incongruente	Incongruente	Verbal

✓ 7- ¿Qué atributos componen los estímulos?

Los atributos que cada estímulo tendrá son: Stroop, Simon, Soporte estimular, Posición en el eje X, Posición en el eje Y, Respuesta correcta, Tipo de efecto Simon. Al respecto:

- De los tres primeros atributos ya se explicó su función y sus variantes.

- La posición de los estímulos será programada mediante ‘vinculación lógica’ de atributos. En las propiedades específicas del objeto de imagen en la diapositiva, ventana ‘Frame’, los campos ‘Position X’ y ‘Position Y’ se completarán con atributos, por ejemplo [ejeX] y [ejeY] respectivamente. En los niveles del objeto List que controla la presentación de estímulos, esos atributos serán completados con ‘valores por ciento’, y sus combinaciones generarán la ubicación de cada una de las cuatro esquinas. Las combinaciones a programar se muestran en la siguiente tabla:

Esquina	Eje X	Eje Y
Izquierda- Arriba	30%	30%
Derecha- Arriba	70%	30%
Izquierda-Abajo	30%	70%
Derecha-Abajo	70%	70%

- La respuesta correcta ante cada estímulo también será programada mediante ‘vinculación lógica’ de atributos. Además, fue añadido un contrabalanceo de teclas por sujeto. La forma de programar este aspecto fue explicada en el apartado 7 de la parte 2.

- El ‘tipo de efecto Simon’ es un atributo que será necesario agregar. El efecto Simon es un conflicto ‘estímulo-respuesta’, que se produce cuando hay cruzamiento entre el hemisferio donde aparece el estímulo y la dirección de la respuesta ejecutada. El contrabalanceo de teclas por sujeto alterna el efecto Simon en los participantes. Concretamente: si para la persona 1 el estímulo aparece en la esquina ‘Derecha-Arriba’ y debe responder con la tecla ‘C’ (tecla de la izquierda), hay incongruencia Simon (hay conflicto). La persona 2 deberá responder al mismo estímulo con la tecla ‘M’ (tecla de la derecha) debido al contrabalanceo, y no habrá incongruencia Simon (no hay cruzamiento entre la ubicación del estímulo y la tecla de respuesta). Entonces, se debe agregar un atributo en el objeto List que controla el contrabalanceo que especifique que: en el nivel 1 de teclas que el efecto Simon es ‘A’, y en el nivel 2 de teclas es el efecto Simon es ‘B’. La matriz del objeto List de contrabalanceo quedará diseñada de la siguiente forma:

TECLA1	TECLA2	SIMONTipo
c	m	A
m	c	B

La matriz del objeto List que controla la presentación de estímulos quedará diseñada como muestra la siguiente tabla (‘vincular lógicamente’ con los atributos del objeto List de contrabalanceo). El atributo ‘Estímulo’ estará escrito en las propiedades específicas del objeto de imagen, en el campo para cargar el archivo externo a EB. La condición Stroop se define por la información que brinda el estímulo y la ubicación en el eje Y. La condición Simon se define por la ubicación del estímulo en el eje X y la tecla de respuesta a ser presionada (debido al contrabalanceo de teclas por sujeto, el efecto Simon cambia de participante a participante).

Stroop	Simon	Soporte	Estímulo	EjeX	EjeY	Correcta	SimonA	SimonB
congr	[SIMONTipo]	palabra	Arriba.jpg	30%	30%	[TECLA1]	congr	incongr
congr	[SIMONTipo]	palabra	Abajo.jpg	30%	70%	[TECLA2]	incongr	congr
incongr	[SIMONTipo]	palabra	Abajo.jpg	70%	30%	[TECLA2]	congr	incongr
incongr	[SIMONTipo]	palabra	Arriba.jpg	70%	70%	[TECLA1]	incongr	congr
incongr	[SIMONTipo]	Iconogr	Down.jpg	30%	30%	[TECLA2]	incongr	congr
incongr	[SIMONTipo]	Iconogr	Up.jpg	30%	70%	[TECLA1]	congr	incongr
congr	[SIMONTipo]	Iconogr	Up.jpg	70%	30%	[TECLA1]	incongr	congr
congr	[SIMONTipo]	Iconogr	Down.jpg	70%	70%	[TECLA2]	congr	incongr
incongr	[SIMONTipo]	palabra	Abajo.jpg	30%	30%	[TECLA2]	incongr	congr
incongr	[SIMONTipo]	palabra	Arriba.jpg	30%	70%	[TECLA1]	congr	incongr
congr	[SIMONTipo]	palabra	Arriba.jpg	70%	30%	[TECLA1]	incongr	congr
congr	[SIMONTipo]	palabra	Abajo.jpg	70%	70%	[TECLA2]	congr	incongr
congr	[SIMONTipo]	Iconogr	Up.jpg	30%	30%	[TECLA1]	congr	incongr
congr	[SIMONTipo]	Iconogr	Down.jpg	30%	70%	[TECLA2]	incongr	congr
incongr	[SIMONTipo]	Iconogr	Down.jpg	70%	30%	[TECLA2]	congr	incongr
incongr	[SIMONTipo]	Iconogr	Up.jpg	70%	70%	[TECLA1]	incongr	congr

✓ 8- ¿Qué tipos de respuesta quiero recoger?

Los datos principales que se recolectarán por cada respuesta serán el TR y el ACC. Adicionalmente, se recolectarán la tecla presionada y cuál era la respuesta correcta en cada estímulo. En EB, este aspecto se programa desde las propiedades del objeto diapositiva, ventana ‘Duración/entrada de datos’, campo ‘Data Logging’, seleccionando el formato de recolección ‘Standard’.

✓ 9- ¿Cómo haré para que las instrucciones sean comprendidas con claridad?

Este aspecto fue trabajado en profundidad en la parte 1 de la guía. Para el presente experimento, se le debe explicar a la persona:

- en el siguiente experimento aparecerán en la pantalla flechas y palabras que indican el sentido ARRIBA o ABAJO.
- Si la flecha o palabra indica ARRIBA, deberá presionar la 'TECLA1'.
- Si la flecha o palabra indica ABAJO, deberá presionar la 'TECLA2'.
- En el medio de la pantalla aparecerá un punto de fijación (un signo +), al que se deberá mirar fijamente durante la tarea. Este punto de fijación se presentará antes y después de cada estímulo (ver secuencia de eventos del experimento en el apartado 7 de la parte 1 de la guía).
- Es importante que responda lo más rápido que pueda y de la mejor manera posible.
- Primero realizará un bloque de ensayos de práctica previamente a realizar la tarea y luego comenzarán los 'ensayos experimentales'.

✓ 10- ¿El mapping será sólo lectura de instrucciones o tendrá mi propia ayuda?

Como se explicó en la parte 1, es recomendable que el mapping sea sólo de lectura de instrucciones. Se presentará un objeto de texto con las instrucciones descritas en el ítem anterior, que en sus propiedades esté programado que tenga una duración infinita y que al presionar la barra espaciadora puede comenzar la tarea.

✓ 11- ¿Necesito presentar un bloque de ensayos previos para asegurar que se comprendió la consigna?

En esta tarea si es necesario presentar un bloque de ensayos previos. La tarea es novedosa para la persona y debe asegurarse un correcto apareamiento entre el estímulo y la tecla de respuesta que le corresponde. El bloque de ensayos de práctica será programado de idéntica manera a los bloques experimentales, pero sólo se presentará a la persona 16 eventos (una vez cada nivel del objeto List).

✓ 12- La presentación: ¿Implica diferentes bloques?

Si. Se utilizará un bloque de práctica y 8 bloques experimentales. Se agregará un objeto List que controle el objeto de List de presentación de estímulos, y en este nuevo objeto List el 'Weight' del procedimiento será igual a 8. Esto indicará a EB que el procedimiento de presentación de estímulos se repita durante 8 veces, logrando presentar 8 bloques experimentales.

En el procedimiento donde se ubica el objeto List que controla la presentación de estímulos, se colocará en una ubicación temporal anterior un objeto de texto con la frase 'Presiona la barra espaciadora cuando estés listo/a para continuar', con una duración infinita, sin recolección de datos, y con la barra espaciadora solamente como tecla permitida. De esta forma, cada vez que termine un bloque experimental, este objeto de texto aparecerá en la pantalla a modo de 'intervalo' entre bloques.

✓ 13- ¿Debería darle feedback a las personas? Si debería hacerlo ¿Será durante todo el proceso?

Se brindará feedback a las personas durante el bloque de práctica y los bloques experimentales. Se debe añadir un objeto de feedback en una ubicación temporal posterior al objeto de diapositiva que presenta los estímulos. Sólo debe permanecer en la 'hoja' de 'respuesta incorrecta' de la diapositiva de feedback un objeto de sonido que presente un 'pitido' cada vez que la persona responda incorrectamente. El resto de los elementos del objeto feedback debe ser eliminado de la diapositiva (devolución de TR y ACC). Concretamente: la persona sólo escuchará un 'pitido' a modo de alarma cada vez que responda incorrectamente.

✓ 14- ¿Cuántos eventos quisiera presentar?

En el bloque de práctica se presentarán 16 eventos para que la persona reciba un entrenamiento previo. Cada bloque experimental estará compuesto de 32 eventos, lo que será un total de 256 ensayos experimentales.

✓ 15- La cantidad de eventos: ¿Resulta lógica en relación a los atributos de los estímulos?

El objeto List que controla la presentación de eventos está compuesto por 16 niveles, y cada nivel se presenta 2 veces ('Weight' = 2) en cada bloque. Como el experimento está compuesto de 8 bloques experimentales, cada evento aparecerá 16 veces en total durante el experimento. La cantidad de niveles se explica por las diferentes combinaciones entre los atributos Stroop-Simon-Soprote estimular-Posición de la imagen.

✓ 16- Los eventos: ¿Tienen un tiempo límite de presentación?

Si. Cada evento puede durar como máximo 10600 ms, y dependerá de la respuesta de la persona. La secuencia del evento se puede observar en la figura del apartado 7 de la parte 1 de la guía, y su duración se descompone de la siguiente forma:

- Un objeto de diapositiva que presente un punto de fijación durante 100 ms, sin que recoleccione una respuesta y que no tenga ninguna tecla permitida para presionar.
- Un objeto de diapositiva que presente el estímulo (imagen de flecha o de palabra) durante 10000 ms ó hasta que la persona emita la respuesta.
- Un objeto de feedback que en todas las 'hojas' de la diapositiva presente el punto de fijación durante 500 ms, y que además en la 'hoja' de respuesta incorrecta tenga incorporado el objeto de sonido de 'pitido'.

✓ 17- ¿La presentación de los eventos será aleatoria?

Si, tanto en el bloque de práctica como en los bloques experimentales. Esto se configura desde las propiedades del objeto List que controla la presentación de los estímulos. Se elige el método de selección aleatorizado.

✓ 18- ¿Es necesario presentar un tipo de contrabalanceo dentro del experimento?

Si. Las teclas de respuesta estarán contrabalanceadas por sujeto, para controlar la velocidad 'típica' de respuesta de cada persona (las personas solemos responder de forma más rápida con una mano o la otra: el contrabalanceo de teclas controla este aspecto de la respuesta). Esto se programa con un objeto List que controle el contrabalanceo, desde el inicio del experimento.

✓ 19- ¿Cuál es la estructura de los estímulos que se van a presentar?

Serán flechas o palabras que indiquen el sentido 'ARRIBA' o 'ABAJO', desde cuatro posiciones posibles en la pantalla.

✓ 20- ¿Qué atributos explícitos conforman la estructura de los estímulos?

Los atributos necesarios para controlar la presentación de los estímulos son: 'Estimulo' (carga el archivo de imagen del estímulo), 'Correcta' (controla la tecla de respuesta), 'ejeX' y 'ejeY' (controla la ubicación de los estímulos).

✓ 21- ¿Cuáles son los atributos implícitos que tendría que incluir para mejorar el análisis de datos?

Se agrega la congruencia o incongruencia 'Stroop' y 'Simon'. Además, se agrega cuando se produce el efecto Simon según el contrabalanceo de teclas 'SIMONTipo' ('SimonA', 'SimonB').

✓ 22- ¿Qué tipo de diseño de programación es el más adecuado?

Para este experimento, se utiliza un diseño cross tasking. En la matriz del objeto List se puede observar el diseño programado.

✓ 23- ¿Por qué una persona sería seleccionada para participar del experimento?

Esto dependerá de las características de la muestra. Controlando la edad de los participantes (por ejemplo, entre 18 y 30 años), cantidad de años de estudio y equilibrando la muestra en género, la muestra será homogénea.

✓ 24- Las personas participantes: ¿Serán expuestas a tratamientos que se supone tendrán efectos en el rendimiento en el experimento?

No es necesario aplicar un tratamiento para realizar la tarea.

✓ 25- ¿Hay otras variables que debería medir para aumentar la validez del experimento?

Para este experimento no es necesario controlar alguna variable adicional.

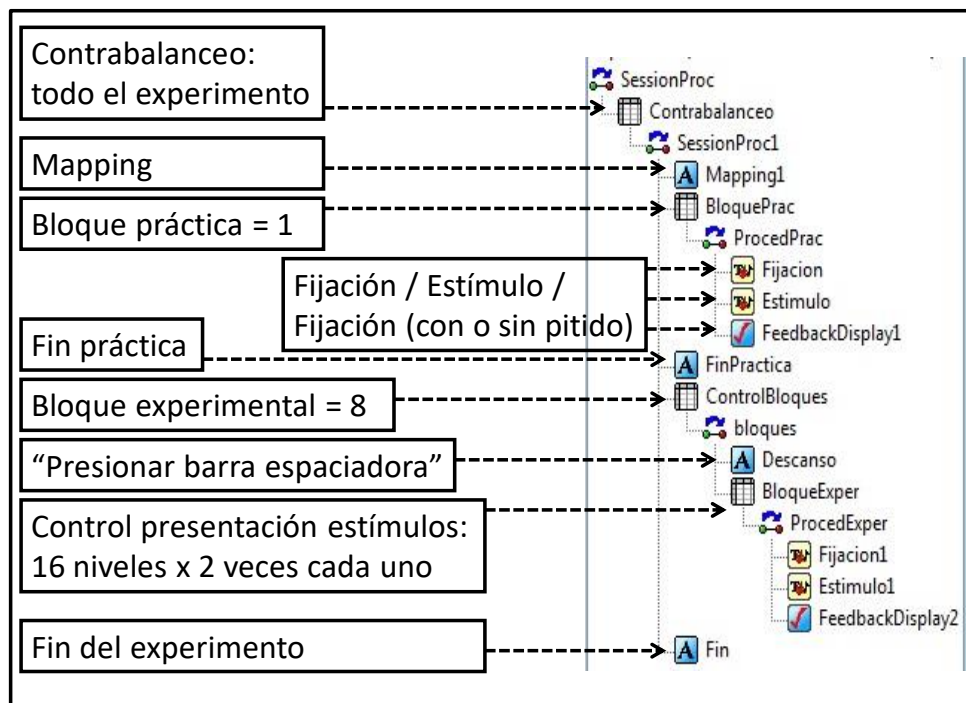
✓ 26- ¿Cómo se les pedirá el 'consentimiento informado'?

El consentimiento informado quedará registrado por escrito. Se debe entregar un formulario de consentimiento informado individual que informe a la persona el título del estudio, objetivos, duración, riesgos e implicaciones, y el equipo de trabajo que lo conforma. Se aclarará que tanto su participación como su retiro del experimento es totalmente voluntario. La persona firmará el consentimiento si está de acuerdo con participar del experimento.

✓ 27- ¿Hay alguna cuestión ética que deba tener especialmente en cuenta?

La realización de la tarea es sencilla: solamente es necesario presionar teclas ante unos estímulos (flechas y palabras) durante unos minutos. No implica riesgo alguno para la persona. Se tendrá en cuenta las consideraciones éticas mínimas contempladas para la investigación en humanos.

La estructura final del experimento debe quedar como muestra la siguiente figura:



Referencias Parte 2

- Botvinick, M. M., Braver, T. S., Barch, D. M., Carter, C. S. & Cohen, J. D. (2001). Conflict Monitoring and Cognitive Control. *Psychological Review*, 108(3), 624-652.
- Botvinick, M. M., Cohen, J. D. & Carter, C. S. (2004). Conflict monitoring and anterior cingulate cortex: an update. *Trends in Cognitive Sciences*, 8(12), 539-546.

Hommel, B. (2010). The Simon effect as tool and heuristic. *Acta Psychologica*, 136(2), 189-202.

Posner, M. I., & Petersen, S. E. (1989). *The attention system of the human brain*. DTIC Document.