

Tarea N°2 - ELO 329 2018-1:
Descuentos, Recomendaciones y Aspectos
Destacados

Pilar Arancibia - Jesús Márquez

Índice

I	Recomendaciones Generales	2
	I-A Respecto al Código de la Tarea	2
	I-B Respecto a la Entrega y Documentación de la Tarea	2
II	Grupo 1 : 201630019-8 201630042-2 201630016-3	3
	II-A Descuentos	3
	II-B Recomendaciones	3
	II-C Aspectos Destacados	3
III	Grupo 2 : 201630006-6 201630039-2 201630002-3	4
	III-A Descuentos	4
	III-B Recomendaciones	4
	III-C Aspectos Destacados	4
IV	Grupo 3 : 201103007-9	4
	IV-A Descuentos	4
	IV-B Recomendaciones	5
V	Grupo 4 : 201573070-9 201890059-1 201790233-7	5
	V-A Descuentos	5
	V-B Recomendaciones	5
	V-C Aspectos Destacados	5
VI	Grupo 5 : 201273065-1 201790232-9 201430022-0	6
	VI-A Descuentos	6
	VI-B Recomendaciones	6
VII	Grupo 6 : 201330015-4 201421003-5 201321074-0	6
	VII-A Descuentos	6
	VII-B Recomendaciones	6
	VII-C Aspectos Destacados	6
VIII	Grupo 7 : 201630012-0 201630015-5 201630004-K	7
	VIII-A Descuentos	7
	VIII-B Recomendaciones	7
	VIII-C Aspectos Destacados	7
IX	Grupo 8 : 201303044-0 201303041-6 201104505-K	7
	IX-A Descuentos	7
	IX-B Recomendaciones	7
	IX-C Aspectos Destacados	8
X	Grupo 9 : 201321034-1 201321007-4 201321056-2	8
	X-A Descuentos	8
	X-B Recomendaciones	8
	X-C Aspectos Destacados	8
XI	Grupo 10 : 201421030-2 201321019-8 201404136-5	8
	XI-A Descuentos	8
	XI-B Recomendaciones	8
	XI-C Aspectos Destacados	8
XII	Grupo 11 : 201321043-0 201730038-8 201704005-K	9
	XII-A Descuentos	9
	XII-B Recomendaciones	9
XIII	Grupo 12 : 201630003-1 201430028-K 201530008-9 201330057-K	9
	XIII-A Descuentos	9

	XIII-B	Recomendaciones	9
	XIII-C	Aspectos Destacados	9
XIV	Grupo 13 : 201504126-1 201504100-8 201504012-5		10
	XIV-A	Descuentos	10
	XIV-B	Recomendaciones	10
	XIV-C	Aspectos Destacados	10
XV	Grupo 14 : 201103032-K 201604009-9 201204122-8		10
	XV-A	Descuentos	10
	XV-B	Recomendaciones	10
	XV-C	Aspectos Destacados	10
XVI	Grupo 15 : 201321009-0 201321051-1 201304179-5		10
	XVI-A	Descuentos	10
	XVI-B	Recomendaciones	11
	XVI-C	Aspectos Destacados	11
XVII	Grupo 16 : 201530022-4 201530025-9 201530014-3		11
	XVII-A	Descuentos	11
	XVII-B	Recomendaciones	11
XVIII	Grupo 17 : 201121020-4 201121032-8		11
	XVIII-A	Descuentos	11
	XVIII-B	Recomendaciones	12
	XVIII-C	Aspectos Destacados	12
XIX	Grupo 18 : 201530013-5 201630005-8 201621058-K		12
	XIX-A	Descuentos	12
	XIX-B	Recomendaciones	13
XX	Grupo 19 : 201030013-7 201030003-K		13
	XX-A	Descuentos	13
	XX-B	Recomendaciones	13
	XX-C	Aspectos Destacados	13
XXI	Grupo 20 : 201430040-9 201430035-2 201430052-2		13
	XXI-A	Descuentos	13
	XXI-B	Recomendaciones	14
XXII	Grupo 21 : 201604062-5 201630034-1		14
	XXII-A	Descuentos	14
	XXII-B	Recomendaciones	14
XXIII	Grupo 22 : 201890108-3 201123008-6 201351025-6		14
	XXIII-A	Descuentos	14

I. RECOMENDACIONES GENERALES

I-A. *Respecto al Código de la Tarea*

- Recordar que para esta tarea se está haciendo uso del concepto de **Modelo-Vista-Controlar**, por lo que era de vital importancia separar el Robot de su vista, así como se separa de su controlador (Pilot), creando la clase **RobotView**. Esto se hace por varias razones, especialmente por eficiencia y el hecho de poder heredar y crear distintas vistas del robot.
- **Es muy importante controlar las entradas del usuario en nuestro código.** Si yo le estoy pidiendo al usuario ingresar un tiempo o una velocidad, el programa no debe de aceptar que se ingresen strings. También es necesario limitar las entradas dependiendo de la capacidad del programa, como por ejemplo, un robot con dirección diagonal hará que la lógica no funcione como se espera, además para velocidades muy grandes también pueden haber complicaciones.
- **Probar para distintos casos de entrada siempre.** Si se le está dando la libertad al usuario de elegir un laberinto o opciones del robot, el programa debiera funcionar para cualquier caso y no solo los dados de ejemplo por el profesor.
- **El programa debe ser auto explicativo respecto a su uso.** Al momento de hacer un programa ustedes deben tener en cuenta que, en el mundo laboral, se trabaja para clientes y usuarios, que casi la totalidad de las veces tiene un nulo conocimiento sobre la programación y ciertas lógicas, por lo que tanto la ejecución, como el uso del programa debe ser lo más claro y fácil de usar posible. Recordar que el README cumple con la función de explicar como ejecutar el programa solamente.
- **Siempre, en cualquier trabajo de programación, se debe de tener un completo control de los errores y excepciones que puedan ocurrir.** Por eso es necesario que para cada trabajo ustedes intenten probar diferentes tipos de entradas y situaciones en las que su código pueda fallar, para poder abordar y arreglar esos fallos.
- **Cuidar el orden de los archivos fuente.** El programa que ustedes confeccionen va a constar de varios archivos, los cuales deben ser entregados a un usuario, a un supervisor o a un compañero, y estos deben ser fáciles de manejar, por lo que es muy importante el **orden**. Utilizar carpetas, separar código de archivos multimedia en carpetas diferentes y con nombres descriptivos.
- **Es importante también tener un orden en nuestro código.** Modulación, Tabulación y eliminación de código inutilizable.
- **El código debe ser auto descriptivo.** En el mundo laboral, se trabaja con códigos que son manipulados por varias personas. Por lo que es de vital importancia, no sólo el orden dentro del código, sino también **los nombres de las variables, clases, atributos y métodos tienen que ser descriptivos y entendibles por sí mismos.** Así cualquier persona que continúe un trabajo pueda ser capaz de entender rápidamente el sentido de cada cosa dentro del código y no perder tiempo en ello.
- **Seguir las especificaciones de entrega es vital.** Si un cliente o un empleador pide un formato de entrega específico (Tanto respecto al formato de archivos fuente, como lo que se pide que haga el programa), el no cumplimiento de esta regla puede significar la pérdida inmediata de un cliente o incluso un despido.
- **El uso de herramientas de manejo de versiones es muy importante.** Es algo que se pide mucho en el ámbito laboral, pues es la forma en que tiene una empresa de centralizar y manejar un programa muy grande que está siendo manejado por muchas personas. No solo es importante saber manejar una herramienta así (como GIT), también es vital el orden y correcto uso de la herramienta. Es importante una descripción de commit coherente e informativa, commits relevantes, cuidado de uso de branch, etc.

I-B. *Respecto a la Entrega y Documentación de la Tarea*

- Recuerden agregar a los ayudantes a sus repositorios y seguir el formato de nombre del repositorio, si no lo hacen sufrirán descuentos y además causarán retrasos en el proceso de revisión de las tareas.
- Hubieron 2 tendencias claras, grupos que usaron muy bien el git y otros que no lo usaron para nada bien. Se recuerda que se descuenta por su uso y que de por sí, una aplicación adecuada de esta herramienta les puede ayudar a la resolución de las tareas de forma más veloz. Recuerden que en la plataforma informativa de tarea están los elementos a evaluar y tutoriales para el uso de GIT.
- **RECORDAR REALIZAR EL TAG FINAL.** Esta opción de los commits permite 'marcar' uno de ellos, siendo ideal para indicar el final de las modificaciones de una entrega. Es por ello que si no lo realizan, es como si no indicaran el fin de su trabajo. Por eso no se considerarán entregadas las tareas que no tengan este elemento presente.
- El readme debe ser breve y conciso, basta con explicar como se compila y ejecuta su tarea, no es necesario listar las clases de cada etapa ni copiar el enunciado de la tarea en ella.
- Cuando copien el formato del makefile en las distintas etapas, recuerden actualizar los macros pertinentes, en especial la clase con el método *main*.
- Ya no se evaluará más JavaDoc, pero recordar que la idea de su uso es el de rotular los parámetros y retornos de los métodos con el fin de mejorar la comprensión de éstos, por ende, deben asegurarse de señalar sus respectivas descripciones de forma simple y breve. Además, recordar de utilizar las opciones del compilador para asegurar la creación de documentación para clases internas y privadas.

- Lean bien los enunciados de las tareas, ya que esto les puede ahorrar trabajo. En esta tarea sólo se exigió el uso de JavaDoc para las clases **Robot y RobotView SÓLO EN SU ÚLTIMA ETAPA DESARROLLADA**. No era necesario la utilización de los rótulos en el resto de las clases ni en cada una de las etapas, fenómeno que se dio bastante.
- Estén atento a los correos provenientes de la lista oficial del ramo, dado que ahí se comunica toda la información relevante del ramo. Ejemplo de ello fue el cambio en la evaluación de la documentación, dado que se paso a exigir diagramas de clases y explicación de cada etapa a un diagrama de clase y una explicación general de la **SU ÚLTIMA ETAPA DESARROLLADA**.
- La explicación de la última etapa desarrollada debería describir brevemente como se relacionan las clases principales de ésta y no listar cada una de las clases utilizadas y mencionar su función principal.
- Recuerden que además de mencionar 3 dificultades que se encontraron al desarrollar la tarea, se pide mencionar como lograron solucionar dicho obstáculo o, en el caso de no lograr solucionarlo, explicar que creen que podría haberles ayudado a abordar mejor el problema.
- Recuerden que el archivo de documentación debe contener el diagrama de clase y la explicación de su última etapa desarrollada y las dificultades y soluciones de su tarea. La idea de éstos ítems es dar contexto de la tarea y su solución de forma breve y concisa.
- Mantener el formato de las etapas. Agrupar cada etapa en una carpeta diferente, incluyendo la etapa extra.
- Sólo entregar archivos fuentes y documentación.

II. GRUPO 1 : 201630019-8 201630042-2 201630016-3

II-A. Descuentos

- **201630016-3** no realiza ningún commit (-8 Puntos en GIT).
- No se utilizan rótulos en clase Robot. (-2 PUNTOS en Uso JavaDoc)
- No se utilizan rótulos en clase RobotView. (-2 PUNTOS en Uso JavaDoc)
- Solo se pueden buscar archivos que se encuentren en la carpeta local. (-2 PUNTO en Etapa 1)
- El tamaño de la ventana no se ajusta al tamaño del laberinto. (-3 PUNTO en Etapa 1)
- Al setear Delta t siempre tira un **NullPointerException** debido a que la clase MyWorld contiene siempre un robot **null**. (-4 PUNTOS en Etapa 2)
- Robot puede ser colocado sobre una pared. (-2 PUNTOS en Etapa 2)
- No se actualiza el listado ni la vista de los robots al momento de abrir otro laberinto. (-2 PUNTOS en Etapa 2)
- No se implementa funcionamiento para estado de play y para estado de pause. (-5 PUNTOS en Etapa 3)
- No se puede crear más de un robot. (-2 PUNTOS en Etapa 3)
- Deja colocar la dirección, pero no el valor de la velocidad. (-1 PUNTO en Etapa 3)
- El robot no posee un movimiento lógico en el laberinto, da solo un salto en el laberinto pudiendo saltar paredes y activando o no los sensores independiente de las paredes cuando está en simulación. (-2 PUNTO en Etapa 3)
- No se presenta Etapa 4, por ende, no se puede evaluar. (-10 PUNTOS en Etapa 4)

II-B. Recomendaciones

- En la documentación no es necesario describir los métodos, eso dejarlo para la documentación generada con javadoc.
- Para la documentación con javadoc se pedía documentar las clases Robot y RobotView, la idea era explicar la clase y cada uno de sus métodos utilizando rótulos para la definición de argumentos y retornos.
- Para esta tarea se pedía el uso del concepto de Modelo-Vista-Controlador, por ello era importante separar el Robot de su vista, creando una clase RobotView.
- Cuidado con no manejar los inputs como en delta t, no deben dejar que el usuario pueda colocar valores inconsistentes, como caracteres, strings, o valores muy grandes.
- Fijarse bien que el repaint() se esté llamando inmediatamente después de la actualización de los componentes gráficos y de forma constante (al final de ActionPerformed).
- Leer bien las especificaciones del profesor con respecto a lo que se pide para la tarea.
- Probar para distintos casos siempre, si se está dejando al usuario elegir un laberinto significa que el programa debiera funcionar igual para cualquier tipo de laberinto y no solo el entregado como ejemplo.

II-C. Aspectos Destacados

- Excelente README y documentación muy profesional de su parte haberlo hecho muy completo y todo en inglés, felicitaciones! (+2 PUNTOS en Bonificaciones varias).

III. GRUPO 2 : 201630006-6 201630039-2 201630002-3

III-A. Descuentos

- No se utilizan rótulos en clase Robot. (-2 PUNTOS en Uso JavaDoc)
- No se utilizan rótulos en clase RobotView. (-2 PUNTOS en Uso JavaDoc)
- Se mencionan una dificultad no solucionada, pero no lo que les hubiera ayudado a resolverla. (-1 PUNTO en Dificultades y Soluciones)
- Se debe seleccionar 2 veces archivo porque la ventana para seleccionar el *maze* se abre 2 veces antes de mostrar el laberinto. (-1 PUNTO en Etapa 1)
- El laberinto seleccionado no se almacena en ninguna parte. (-3 PUNTOS en Etapa 1)
- Primer boton de pausa necesita 2 click para alternar a play. (-1 PUNTO en Etapa 1)
- Se muestra ventana para entregar valor del delta, pero no realiza cambio alguno en objeto world. (-3 PUNTOS en Etapa 2)
- El objeto robot no es generado ni almacenado en ningún momento, por lo que tampoco se fijan valores predeterminados para la posición y la velocidad. (-5 PUNTOS en Etapa 2)
- Es posible crear un robot sin que haya un maze seleccionado, lo que genera una excepción *NullPointerException*. (-3 PUNTOS en Etapa 2)
- No se maneja adecuadamente el caso en que no se ingrese un número en el menú de velocidad. (-1 PUNTO en Etapa 3)
- No existe menú para definir piloto. (-2 PUNTOS en Etapa 3)
- No se selecciona la ubicación del robot con el mouse. (-5 PUNTOS en Etapa 3)
- No se implementa funcionamiento para estado de play y para estado de pause. (-5 PUNTOS en Etapa 3)
- No se puede crear más de un robot. (-2 PUNTOS en Etapa 3)
- Los sensores no cambian de color. (-2 PUNTOS en Etapa 3)
- El robot no se mueve en el laberinto. (-2 PUNTOS en Etapa 3)
- No se presenta Etapa 4, por ende, no se puede evaluar. (-10 PUNTOS en Etapa 4)
- En la entrega se incorporan documentación generada por Javadoc en cada una de las etapas, lo que es considerado archivos basura en la entrega. (1 PUNTO a Descuentos Varios)
- En la etapa 2 y 3 ya no se permite seleccionar un *maze* que no sea el entregado por el grupo como ejemplo. (1 PUNTO a Descuentos Varios)

III-B. Recomendaciones

- Recordar que se trabaja con arquitectura modelo-vista-controlador, por lo que no se puede dejar de lado uno de estos módulos y no darles funcionalidad, porque la interfaz resultante no operará como se desea.
- Comentar de mejor manera el código, tratando de realizar comentarios sobre pequeñas secciones de código y no solo una descripción general de cada método y/o clase.

III-C. Aspectos Destacados

- Muy buen uso de GIT. Felicitaciones y sigan así.
- Buen y divertido detalle el de seleccionar un icono para la vista en la barra de tareas del programa.

IV. GRUPO 3 : 201103007-9

IV-A. Descuentos

- Para la tarea pasada se dio aviso de descuento por no respetar el formato de carpetas y aún así no se respetó (Por más tarde que se haya dado el aviso todo es conversable con respecto al tiempo y no demora mucho tiempo el cambio de formato y makefiles). (-5 PUNTOS en Descuentos)
- Archivos basura y que no corresponden al proyecto (Archivos de tarea 1). (-2 PUNTOS en Descuentos)
- La documentación corresponde a la misma entrega para la tarea 1. (-16 PUNTOS en Documentación)
- Mismo README de tarea 1, explica como compilar y ejecutar dicha tarea. (-1 PUNTO a README)
- No se hace uso de JavaDoc. (-4 PUNTOS en Uso de JavaDoc)
- Exactamente mismo makefile de la tarea 1. (-1 PUNTO en Makefile)
- Se entiende que la etapa 1 es contenida en el zip con el nombre de la etapa, pero al ejecutar el código no muestra ninguna ventana, no hace uso de paneles, ni obedece a lo pedido por el profesor. (-15 PUNTOS en Etapa 1)
- Se entiende que la etapa 2 es contenida en el zip con el nombre de la etapa, pero al ejecutar el código no muestra ninguna ventana, no hace uso de paneles, ni obedece a lo pedido por el profesor. (-15 PUNTOS en Etapa 2)

- Se entiende que la etapa 3 es contenida en la carpeta src, pero la clase RobotView está mal implementada y está actuando como el panel cuando debiera representar solo la vista del robot. Además tampoco puede ser ejecutada, debido a que lanza inmediatamente una excepción **IllegalComponentStateException**, debido a que el panel está quedando en null. **(-20 en Etapa 3)**
- No se presenta Etapa 4, por ende, no se puede evaluar. **(-10 PUNTOS en Etapa 4)**

IV-B. *Recomendaciones*

- A pesar de que al principio se dejó pasar el Disclaimer debido a que la idea de las carpetas era contribuir al orden y su método también funcionaba, tienen que comprender que en el mundo laboral si un cliente o el supervisor les pide un formato de entrega y ustedes no las respetan, eso significa o despido o perder el cliente. Nosotros como ayudantes pedimos la separación de etapas por carpetas porque se nos hace más fácil y genérica la revisión de tareas, por lo que cosas así nos retrasan en la revisión y no nos aportan.
- Los archivos png y demás que se encuentran en la primera impresión del proyecto se ven muy desordenados, se debió de guardar y administrar cada uno de esos archivos en carpetas correspondientes y con nombres descriptivos.
- Se debe crear un nuevo repositorio limpio para cada tarea.
- No utilizar herramientas de otros programas como **Netbeans** (Archivos con extensión form) sin mencionarlo en el README y sin tener un método con makefiles para correrlos, pues no se tomarán en cuenta, ya que se utiliza Aragon para revisar las tareas y si el makefile no es capaz de reproducir estos archivos, entonces se toman como archivos basura.
- Tanto el profesor como los ayudantes pedimos una estructura y un formato de código y entrega, la cual debe ser respetada, pues para la tercera tarea ya no se recibirán proyectos que no cumplan el formato.
- Es de completa responsabilidad del alumno/Programador que el Ayudante/Cliente sea capaz de entender y ejecutar su código de forma fácil y rápida, tengan en cuenta que los usuarios y clientes son, en su gran mayoría, gente inentendida del tema del ramo y de cualquier tipo de lógica programática.

V. GRUPO 4 : 201573070-9 201890059-1 201790233-7

V-A. *Descuentos*

- Mensajes de los commits son poco informativos, Muchos commits poco relevante (cambian una sola línea o cambios con poco impacto), Muchos commits tienen el mismo nombre. **(-2 PUNTOS a Uso de Repositorio GIT)**
- No se utilizan rótulos en clase Robot. **(-2 PUNTOS en Uso JavaDoc)**
- No se utilizan rótulos en clase RobotView. **(-2 PUNTOS en Uso JavaDoc)**
- No se crea una ventana para redefinir el valor de delta t. **(-5 PUNTOS en Etapa 2)**
- No se actualiza el listado ni la vista de los robots al momento de abrir otro laberinto. **(-2 PUNTOS en Etapa 2)**
- La vista del robot no se ajusta para que se encuentre dentro del laberinto a utilizar. **(-2 PUNTOS en Etapa 2)**
- No se maneja adecuadamente el caso en que no se ingrese un número en el menú de velocidad. **(-1 PUNTO en Etapa 3)**
- No se muestran ambas opciones exigidas de piloto. **(-1 PUNTO en Etapa 3)**
- No se presenta Etapa 4, por ende, no se puede evaluar. **(-10 PUNTOS en Etapa 4)**
- En la entrega se incorporan documentación generada por Javadoc, lo que es considerado archivos basura en la entrega. **(1 PUNTO a Descuentos Varios)**

V-B. *Recomendaciones*

- No es necesario para proyectos pequeños como las tareas mencionar una a una las clases generadas en cada etapa.
- Para reducir tiempo de redacción, recuerden que solo se exige la explicación de las relaciones entre las clases para la última etapa desarrollada.
- Centrar las imágenes colocadas en el pdf de documentación.
- Tratar de abordar situaciones que podrían corromper el funcionamiento del programa, como colocar por error una letra en vez de un número para el delta t.
- Comentar de mejor manera el código, tratando de realizar comentarios sobre pequeñas secciones de código y no solo una descripción general de cada método y/o clase.

V-C. *Aspectos Destacados*

- Muy buena redacción, agradable a la lectura.
- Buen formato utilizado en el pdf de documentación.

VI. GRUPO 5 : 201273065-1 201790232-9 201430022-0

VI-A. Descuentos

- **201430022-0** Realiza menos de 4 commits, Sube etapas completas (-4 PUNTOS en GIT).
- **201790232-9** Muchos commits tienen el mismo nombre, Mensajes de los commits son poco informativos. (-2 PUNTOS en GIT).
- No se hace uso de JavaDoc para las clases Robot y RobotView (-4 PUNTOS en Uso JavaDoc)
- Etapa 1 tira un error de compilación porque no encuentra la clase Vector2D. (-2 PUNTOS en Compilación)
- El tamaño de la ventana no se ajusta al tamaño del laberinto. (-3 PUNTO en Etapa 1)
- Es posible crear un robot sin que haya un maze seleccionado. (-2 PUNTOS en Etapa 2)
- Robot puede ser colocado sobre una pared. (-2 PUNTOS en Etapa 2)
- No se crea una ventana para redefinir el valor de delta t. (-5 PUNTOS en Etapa 2)
- No se actualiza el listado ni la vista de los robots al momento de abrir otro laberinto. (-2 PUNTOS en Etapa 2)
- No se puede seleccionar la ubicación del robot con el mouse en la carpeta correspondiente a la etapa y al correr el programa de la carpeta Mousse el robot sigue el mouse, pero un sensor queda flotando, puede posicionarse fuera del laberinto o encima de las paredes, a veces deja de seguir el mouse sin ningún motivo, y el programa de esa carpeta no funciona para nada más. (-4 PUNTOS en Etapa 3)
- No se puede crear más de un robot en el programa de la carpeta de la etapa, en Mousse se crea independiente de la simulación, pero dejan de seguir al mouse llegado un punto y luego desaparecen de la ventana. (-2 PUNTOS en Etapa 3)
- No se modifica la velocidad después de ingresarla al menú. (-1 PUNTO en Etapa 3)
- No se modifica el piloto después de ingresarlo al menú. (-1 PUNTO en Etapa 3)
- El movimiento del robot no es lógico, traspasa paredes y los sensores se prenden o se apagan cuando no corresponde. (-2 PUNTOS en Etapa 3)
- No se presenta Etapa 4, por ende, no se puede evaluar. (-10 PUNTOS en Etapa 4)
- No agrega a ambos ayudantes al repositorio. (-2 PUNTOS en Descuentos).
- Múltiples archivos basura, especialmente en etapa 3. (-1 PUNTO en Descuentos).

VI-B. Recomendaciones

- Al momento de elegir el tipo de piloto un usuario no tiene como saber qué tipos de pilotos hay y qué es lo que tengo que escribir para elegir el deseado.
- Cuidado con no manejar los inputs, no deben dejar que el usuario pueda colocar valores inconsistentes, como caracteres, strings, o valores muy grandes.
- Los nombres de los métodos deben ser descriptivos.
- Tener cuidado con los tamaños, tratar de ajustar los tamaños de los objetos dependiendo del tamaño de su entorno, para este caso su robot era demasiado grande en comparación a la brecha entre paredes del laberinto.

VII. GRUPO 6 : 201330015-4 201421003-5 201321074-0

VII-A. Descuentos

- El nombre del repositorio no sigue el formato entregado, algunos de los mensajes de los commits son poco informativos y tienen commits poco relevante (cambian una sola línea, un solo archivo o los cambios tienen poco impacto). (-2 PUNTOS a Uso de Repositorio GIT)
- Algunos laberintos seleccionados no se visualizan por completo. (-2 PUNTOS en Etapa 1)
- Es posible crear un robot sin que haya un maze seleccionado. (-2 PUNTOS en Etapa 2)
- No se actualiza el listado ni la vista de los robots al momento de abrir otro laberinto. (-2 PUNTOS en Etapa 2)
- Robot puede ser colocado sobre una pared. (-2 PUNTOS en Etapa 3)

VII-B. Recomendaciones

- No es necesario realizar 3 dificultades y soluciones por etapa, se piden 3 por la tarea en general.
- Si manejan un error, no es necesario mostrarlo por el terminal, se recomienda almacenarlo en un archivo de log si se considera pertinente.

VII-C. Aspectos Destacados

- Muy buen formato de pdf, agradable a la vista.
- Documentación con mucha dedicación, se nota esfuerzo realizado.

VIII. GRUPO 7 : 201630012-0 201630015-5 201630004-K

VIII-A. Descuentos

- **201630004-K** Muchos commits poco relevantes, Mensajes de los commits son poco informativos (start working on ..) (-3 PUNTOS en GIT)
- El tamaño de la ventana no se ajusta al tamaño del laberinto. (-3 PUNTO en Etapa 1)
- Es posible crear un robot sin que haya un maze seleccionado, lo que genera una excepción *NullPointerException*. (-3 PUNTOS en Etapa 2)
- Robot puede ser colocado sobre una pared. (-2 PUNTOS en Etapa 2)
- No se actualiza el listado ni la vista de los robots al momento de abrir otro laberinto. (-2 PUNTOS en Etapa 2)
- No se crea una ventana para redefinir el valor de delta t. (-5 PUNTOS en Etapa 2)
- Se puede crear un robot aún estando en simulación (-1 PUNTOS en Etapa 3)
- No se presenta Etapa 4, por ende, no se puede evaluar. (-10 PUNTOS en Etapa 4)

VIII-B. Recomendaciones

- Al generar la documentación con make doc es recomendable generar una carpeta en la que se guarden los archivos html por temas de orden, o también hacer que el make clean pueda encargarse de limpiarlos.
- De la misma manera intentar hacer que cualquier archivo que no corresponda a código (archivos multimedia por ejemplo) estén contenidos dentro de una carpeta que los separe del código.
- Hay que tener cuidado con los inputs, si bien ustedes no dejan que el usuario sea capaz de colocar entradas que no son números (que está muy bien!), también es necesario limitar las entradas dependiendo de la capacidad del programa, como por ejemplo, un robot con dirección diagonal hará que la lógica no funcione como se espera, o al colocar las dos velocidades en 0 podemos tener un robot sin dirección, además para velocidades muy grandes también pueden haber complicaciones, tenerlo en cuenta!

VIII-C. Aspectos Destacados

- Excelente README.

IX. GRUPO 8 : 201303044-0 201303041-6 201104505-K

IX-A. Descuentos

- Mensajes de los commits son poco informativos, Muchos commits tienen el mismo mensaje, Hacen múltiples tag Final (en esta ocasión se considerará el último commit, para siguientes, hasta el primer tag final). (-2 PUNTOS a Uso de Repositorio GIT)
- No se actualiza clase objetivo a ejecutar en makefile de etapas 2, 3, y 4. (-1 PUNTO a Makefile)
- Más de la mitad de los parámetros presentes en la clase Robot Generan un Error. (-1 PUNTO en Uso JavaDoc)
- No se genera la documentación de la clase RobotView después de ejecutar el comando JavaDoc. (-2 PUNTOS en Uso JavaDoc)
- Solo se pueden seleccionar archivos que se encuentren en la carpeta local. (-2 PUNTOS en Etapa 1)
- Algunos laberintos seleccionados no se visualizan por completo. (-2 PUNTOS en Etapa 1)
- No se maneja adecuadamente el caso en que no se ingrese un número en el menú de velocidad. (-2 PUNTOS en Etapa 2)
- Es posible crear un robot sin que haya un maze seleccionado, lo que genera una excepción *NullPointerException*. (-3 PUNTOS en Etapa 2)
- No se actualiza el listado ni la vista de los robots al momento de abrir otro laberinto. (-2 PUNTOS en Etapa 2)
- No se maneja adecuadamente el caso en que no se ingrese un número en el menú de velocidad. (-1 PUNTO en Etapa 3)
- Valores seleccionados de velocidad no se reflejan adecuadamente en la simulación. (-1 PUNTO en Etapa 3)

IX-B. Recomendaciones

- Verificar la ubicación de los rótulos al momento de utilizar comentarios tipos javadoc.
- No añadir opciones y menús sin funcionalidad, como en la etapa 1.
- Recordar actualizar adecuadamente el makefile.
- Trabajar mejor las excepciones.

IX-C. Aspectos Destacados

- Buen formato de documentación.
- Excelente redacción.

X. GRUPO 9 : 201321034-1 201321007-4 201321056-2

X-A. Descuentos

- Al abrir un segundo laberinto distinto al anterior, estos se superponen pues no se borra el anterior. **(-2 PUNTO en Etapa 1)**
- No se actualiza el listado ni la vista de los robots al momento de abrir otro laberinto. **(-2 PUNTOS en Etapa 2)**
- Robot puede ser colocado sobre una pared. **(-2 PUNTOS en Etapa 2)**

X-B. Recomendaciones

- Al momento de elegir el tipo de piloto un usuario no tiene como saber qué tipos de pilotos hay y qué es lo que tengo que escribir para elegir el deseado.
- Probar para distintos casos siempre, tratar de romper el código o generar entradas poco convencionales para poder abordar todos los problemas que pudiera llegar a tener el programa y así poder llegar a un código más robusto. Recuerden que los usuarios usualmente son gente sin conocimiento alguno sobre el proyecto o cualquier cosa que tenga que ver con la programación.

X-C. Aspectos Destacados

- Una excelente y muy completa documentación es algo muy profesional y dedicado de su parte, muchas felicitaciones!. **(+2 PUNTOS en Bonificaciones varias)**
- Excelente README.
- Excelente documentación y uso de JavaDoc.
- Muy buen uso de GIT y trabajo en equipo!.
- Excelente el uso de comentarios en el código.
- Un trabajo excelente en todos los aspectos, felicitaciones!.

XI. GRUPO 10 : 201421030-2 201321019-8 201404136-5

XI-A. Descuentos

- No se añade la explicación de como se relacionan entre sí las clases principales de la última etapa desarrollada. **(-7 Puntos en Explicación)**
- Algunos laberintos seleccionados no se visualizan por completo. **(-2 PUNTOS en Etapa 1)**
- Primer boton de play necesita 2 click para alternar a pausa. **(-2 PUNTOS en Etapa 1)**
- Es posible crear un robot sin que haya un maze seleccionado, lo que genera una excepción *NullPointerException*. **(-3 PUNTOS en Etapa 2)**
- No se actualiza el listado ni la vista de los robots al momento de abrir otro laberinto. **(-2 PUNTOS en Etapa 2)**
- La vista del robot no se ajusta para que se encuentre dentro del laberinto a utilizar. **(-2 PUNTOS en Etapa 2)**
- No se maneja adecuadamente el caso en que no se ingrese un número en el menú de velocidad. **(-1 PUNTO en Etapa 3)**
- Valores seleccionados de velocidad no se reflejan adecuadamente en la simulación. **(-1 PUNTO en Etapa 3)**
- Robot puede ser colocado sobre una pared. **(-2 PUNTOS en Etapa 3)**
- No se almacena el tiempo en los datos de salida. **(-2 PUNTOS en Etapa 4)**
- Datos sin encabezado, es difícil saber qué valores son qué y a qué robot pertenecen. **(-1 PUNTO en Etapa 4)**

XI-B. Recomendaciones

- Agregar descripciones a todos los rótulos utilizados.

XI-C. Aspectos Destacados

- Muy buen uso de Git, felicitaciones.

XII. GRUPO 11 : 201321043-0 201730038-8 201704005-K

XII-A. Descuentos

- **201321043-0** Mensajes de los commits son poco informativos, Muchos commits tienen el mismo nombre, Muchos commits poco relevantes. **(-4 PUNTOS en GIT)**
- **201730038-8** Mensajes de los commits son poco informativos, Muchos commits tienen el mismo nombre, lenguaje poco formal en los commits, . **(-3 PUNTOS en GIT)**
- **201704005-K** 1 solo commit, descripción poco informativa, ningún aporte a código. **(-7 PUNTOS en GIT).**
- No se explica la interacción entre los módulos. **(-7 PUNTOS en Documentación)**
- No se hace uso de JavaDoc para las clases Robot y RobotView **(-4 PUNTOS en Uso JavaDoc)**
- Al abrir un segundo laberinto distinto al anterior, estos se superponen pues no se borra el anterior. **(-2 PUNTO en Etapa 1)**
- Se genera un **error** de compilación en la etapa 2, debido a que el archivo .java no tiene el mismo nombre que 1. **(-2 PUNTOS en Compilación)**
- Es posible crear un robot sin que haya un maze seleccionado, lo que genera una excepción *NullPointerException*. **(-3 PUNTOS en Etapa 2)**
- No se crea una ventana para redefinir el valor de delta t. **(-5 PUNTOS en Etapa 2)**
- No se actualiza el listado ni la vista de los robots al momento de abrir otro laberinto. **(-2 PUNTOS en Etapa 2)**
- Robot puede ser colocado sobre una pared. **(-2 PUNTOS en Etapa 2)**
- No se presenta Etapa 3, por ende, no se puede evaluar. **(-20 PUNTOS en Etapa 3)**
- No se presenta Etapa 4, por ende, no se puede evaluar. **(-10 PUNTOS en Etapa 4)**

XII-B. Recomendaciones

- El uso de GIT se encuentra en una instancia formal, la descripción de los commits deben ser informativas respecto al cambio que se hizo y los cambios deben ser relevantes.
- Lean bien las especificaciones pedidas para la tarea.
- Intentar hacer que cualquier archivo que no corresponda a código (archivos multimedia por ejemplo) estén contenidos dentro de una carpeta que los separe del código.

XIII. GRUPO 12 : 201630003-1 201430028-K 201530008-9 201330057-K

XIII-A. Descuentos

- No se genera la documentación de la clase RobotView después de ejecutar el comando JavaDoc. **(-2 PUNTOS en Uso JavaDoc)**
- Solo se pueden seleccionar archivos que se encuentren en la carpeta local. **(-2 PUNTOS en Etapa 1)**
- No se actualiza el listado ni la vista de los robots al momento de abrir otro laberinto. **(-2 PUNTOS en Etapa 2)**
- La vista del robot no se ajusta para que se encuentre dentro del laberinto a utilizar. **(-2 PUNTOS en Etapa 2)**
- Valores seleccionados de velocidad no se reflejan adecuadamente en la simulación. **(-1 PUNTO en Etapa 3)**
- Robot puede ser colocado sobre una pared. **(-2 PUNTOS en Etapa 3)**
- No se despliegan los datos de posición de ninguna forma, por lo que no se determinan si son valores lógicos. **(-5 PUNTOS en Etapa 4)**
- No se crea archivo .txt, .csv u otro afín con la posición de los robot en el tiempo de simulación. **(-5 PUNTOS en Etapa 4)**

XIII-B. Recomendaciones

- La selección usada para el tipo de dato para delta t no es la más conveniente, ya que para simulaciones tiempos enteros podrían no permitir captar determinadas interacciones.

XIII-C. Aspectos Destacados

- Excelente uso de GIT, sigan así.
- Muy buen readme, breve pero conciso.
- Genial archivo de documentación, contiene la información justa y necesaria, usa un formato adecuado y agradable a la vista y el uso de **negritas** para marcar los puntos principales de los párrafos está muy bien ejecutado. Felicitaciones.

XIV. GRUPO 13 : 201504126-1 201504100-8 201504012-5

XIV-A. Descuentos

- No se explica la interacción entre los módulos. (-7 PUNTOS en Documentación)
- No se crea la clase RobotView, por lo que no se encuentra documentada. (-2 PUNTOS en Uso de JavaDoc).
- El tamaño de la ventana no se ajusta al tamaño del laberinto. Es la imagen jpg de un laberinto el que se está ajustando a la ventana. (-3 PUNTO en Etapa 1)
- Es posible acceder a la ventana de cambio de delta sin abrir ningún maze seleccionado. (-1 PUNTO en Etapa 2)
- Se muestra ventana para entregar valor del delta, pero no realiza cambio alguno en objeto world. (-3 PUNTOS en Etapa 2)
- Robot puede ser colocado sobre una pared. (-2 PUNTOS en Etapa 2)
- No se actualiza el listado ni la vista de los robots al momento de abrir otro laberinto. (-2 PUNTOS en Etapa 2)
- No se presenta Etapa 3, por ende, no se puede evaluar. (-20 PUNTOS en Etapa 3)
- No se presenta Etapa 4, por ende, no se puede evaluar. (-10 PUNTOS en Etapa 4)

XIV-B. Recomendaciones

- Para esta tarea se pedía el uso del concepto de Modelo-Vista-Controlador, por ello era importante separar el Robot de su vista, creando una clase RobotView de forma interna o separada.
- Cuidado con no manejar los inputs como en delta t, no deben dejar que el usuario pueda colocar valores inconsistentes, como caracteres, strings, o valores muy grandes.
- Intentar hacer que cualquier archivo que no corresponda a código (archivos multimedia por ejemplo) estén contenidos dentro de una carpeta que los separe del código.

XIV-C. Aspectos Destacados

- Excelente README!.

XV. GRUPO 14 : 201103032-K 201604009-9 201204122-8

XV-A. Descuentos

- Mensajes de los commits son poco informativos, Muchos commits poco relevante (cambian una sola linea, un solo archivo o los cambios tienen poco impacto), Se suben etapas completas, Realizan commits después del tag Final (en esta ocasión se considerará el último commit, para siguientes, hasta el primer tag final). (-4 PUNTOS a Uso de Repositorio GIT)
- No se maneja adecuadamente el caso en que no se ingrese un número en el menú de delta t. (-2 PUNTOS en Etapa 2)
- Es posible crear un robot sin que haya un maze seleccionado. (-2 PUNTOS en Etapa 2)
- No se actualiza el listado ni la vista de los robots al momento de abrir otro laberinto. (-2 PUNTOS en Etapa 2)
- La vista del robot no se ajusta para que se encuentre dentro del laberinto a utilizar. (-2 PUNTOS en Etapa 2)
- Valores seleccionados de velocidad no se reflejan adecuadamente en la simulación. (-1 PUNTO en Etapa 3)
- No se modifica el piloto después de ingresarlo al menú. (-1 PUNTO en Etapa 3)
- Robot puede ser colocado sobre una pared. (-2 PUNTOS en Etapa 3)
- Se puede crear un robot aún estando en simulación (-1 PUNTOS en Etapa 3)
- Datos sin encabezado, es difícil saber qué valores son qué. (-1 PUNTO en Etapa 4)

XV-B. Recomendaciones

- Readme es muy extenso, menciona cada opción disponible en cada etapa y cada una de las clases que lo componen, basta con explicar brevemente el funcionamiento del programa y como compilarlo

XV-C. Aspectos Destacados

- Buena documentación, breve pero concisa.

XVI. GRUPO 15 : 201321009-0 201321051-1 201304179-5

XVI-A. Descuentos

- No se explica la interacción entre los módulos. (-7 PUNTOS en Documentación)
- No se actualiza el listado ni la vista de los robots al momento de abrir otro laberinto. (-2 PUNTOS en Etapa 2)
- Robot puede ser colocado sobre una pared. (-2 PUNTOS en Etapa 2)
- Robots no cuentan con gráfica de sensores que cambien de color. (-2 PUNTOS en Etapa 3)
- Datos sin descripción, es difícil saber qué valores son qué y a qué robot pertenecen. (-2 PUNTOS en Etapa 4)

XVI-B. *Recomendaciones*

- Hay que tener cuidado con los inputs, también es necesario limitar las entradas dependiendo de la capacidad del programa, como por ejemplo, un robot con dirección diagonal hará que la lógica no funcione como se espera, o al colocar las dos velocidades en 0 podemos tener un robot estático, además para velocidades muy grandes también pueden haber complicaciones, tengánlo en cuenta!.
- Para la etapa 4/5 faltó ser más descriptivos al momento de usar la opción de **Record** y **Save routes**, no se entendía a primera vista cómo es que se guardan los valores, qué formato se debe elegir al guardar y cómo interpretar los resultados. Tengan en cuenta que la mayoría de los usuarios son gente inentendida del proyecto y de cualquier tipo de lógica programática.

XVI-C. *Aspectos Destacados*

- Excelente README!.
- Muy buen uso de JavaDoc!.
- Muy buena tarea en general, felicitaciones!.

XVII. GRUPO 16 : 201530022-4 201530025-9 201530014-3

XVII-A. *Descuentos*

- El nombre del repositorio no sigue el formato solicitado, Mensajes de los commits son poco informativos, Se suben etapas completas, 201530025-9 Realiza menos de 4 commits. Para 201530022-4 y 201530014-3 (-2 PUNTOS a Uso de Repositorio GIT), Para 201530025-9 (-6 PUNTOS a Uso de Repositorio GIT).
- Archivo Readme está en blanco. (-1 PUNTO a README)
- No se hace nada cuando se ejecuta make. (-1 PUNTO a Makefile)
- No se utilizan rótulos en clase Robot. (-2 PUNTOS en Uso JavaDoc)
- No se utilizan rótulos en clase RobotView. (-2 PUNTOS en Uso JavaDoc)
- No se agrega diagrama de clases de última etapa desarrollada a archivo de documentación. (-2 PUNTOS en Diagramas de Clases)
- No se realiza una explicación de las clases de la última etapa desarrollada. (-7 PUNTOS en Explicación de Módulos)
- No se mencionan las dificultades encontradas en la tarea ni sus soluciones. (-7 PUNTOS en Dificultades y Soluciones)
- Al abrir un segundo laberinto distinto al anterior, estos se superponen. (-1 PUNTO en Etapa 1)
- Es posible crear un robot sin que haya un maze seleccionado. (-2 PUNTOS en Etapa 2)
- No se actualiza el listado ni la vista de los robots al momento de abrir otro laberinto. (-2 PUNTOS en Etapa 2)
- La vista del robot no se ajusta para que se encuentre dentro del laberinto a utilizar. (-2 PUNTOS en Etapa 2)
- La vista del robot no incluye a los sensores. (-2 PUNTOS en Etapa 2)
- No se modifica la velocidad después de ingresarla al menú. (-1 PUNTO en Etapa 3)
- No se maneja adecuadamente el caso en que no se ingrese un número en el menú de velocidad. (-1 PUNTO en Etapa 3)
- No se modifica el piloto después de ingresarlo al menú. (-1 PUNTO en Etapa 3)
- No se puede seleccionar la ubicación del robot con el mouse. (-5 PUNTOS en Etapa 3)
- No se implementa funcionamiento para estado de play y para estado de pause. (-5 PUNTOS en Etapa 3)
- No se puede crear más de un robot. (-2 PUNTOS en Etapa 3)
- Los sensores no cambian de color porque no fueron incorporados. (-2 PUNTOS en Etapa 3)
- El robot no se mueve en el laberinto. (-2 PUNTOS en Etapa 3)
- No se presenta Etapa 4, por ende, no se puede evaluar. (-10 PUNTOS en Etapa 4)
- Existen múltiples archivos de documentación. (1 PUNTO a Descuentos Varios)
- Se entregan archivos .class en etapa 3, archivos que no corresponden a código fuente. (1 PUNTO a Descuentos Varios)

XVII-B. *Recomendaciones*

- Consultar a Profesores o Ayudantes si tienen dudas sobre la tarea.
- Leer formato de la documentación, para obtener puntaje por desarrollo de ésta.

XVIII. GRUPO 17 : 201121020-4 201121032-8

XVIII-A. *Descuentos*

- No se utilizan rótulos en clase Robot ni en RobotView, además no se genera la documentación con el comando **make doc**. (-4 PUNTOS en Uso JavaDoc)

- La orientación para cualquier laberinto es siempre la misma. (-1 PUNTO en Etapa 1)
- El tamaño de la ventana no se ajusta al tamaño del laberinto. (-3 PUNTO en Etapa 1)
- Es posible crear un robot sin que haya un maze seleccionado, lo que genera una excepción *NullPointerException*. (-3 PUNTOS en Etapa 2)
- Al intentar abrir otro laberinto luego de colocar el robot, tanto el robot como el laberinto desaparecen y no es posible volver a colocarlos en la ventana. (-2 PUNTOS en Etapa 2)
- Robot puede ser colocado sobre una pared. (-2 PUNTOS en Etapa 2)
- No se está agregando el robot al mundo, solo se dibuja en el panel. (-1 PUNTOS en Etapa 2)
- No se crea una ventana para redefinir el valor de delta t. (-5 PUNTOS en Etapa 2)
- No se modifica la velocidad después de ingresarla al menú. (-1 PUNTO en Etapa 3)
- No se modifica el piloto después de ingresarlo al menú. (-1 PUNTO en Etapa 3)
- No se puede crear más de un robot. (-2 PUNTOS en Etapa 3)
- Los sensores frontal e izquierdo no cambian de color. (-2 PUNTOS en Etapa 3)
- Robot no sigue al mouse, solo aparece luego de presionar el botón izquierdo. (-2 PUNTOS en Etapa 3)
- El robot setea bien su dirección sólo cuando es colocado fuera del laberinto, y si es posicionado varias veces el laberinto desaparece del panel. (-2 PUNTOS en Etapa 3)
- No se presenta Etapa 4, por ende, no se puede evaluar. (-10 PUNTOS en Etapa 4)

XVIII-B. Recomendaciones

- OJO, si están haciendo uso de los makefiles para compilar y/o ejecutar su código, tienen que especificar en el README que esa es la forma de compilar y/o ejecutar su código y no colocar solamente la forma general de hacerlo.
- Probar para distintos casos siempre, tratar de romper el código o generar entradas poco convencionales para poder abordar todos los problemas que pudiera llegar a tener el programa y así poder llegar a un código más robusto. Recuerden que los usuarios usualmente son gente sin conocimiento alguno sobre el proyecto o cualquier cosa que tenga que ver con la lógica programática.

XVIII-C. Aspectos Destacados

- Se nota un buen trabajo en equipo!.

XIX. GRUPO 18 : 201530013-5 201630005-8 201621058-K

XIX-A. Descuentos

- Se suben etapas completas, hacen múltiples tag Final (en esta ocasión se considerará el último commit, para siguientes, hasta el primer tag final), 201530013-5 no realiza commits. Para 201630005-8 y 201621058-K (-2 PUNTOS a Uso de Repositorio GIT), para 201530013-5 (-8 PUNTOS a Uso de Repositorio GIT)
- No hay makefiles para las etapas 1 y 2. (-1 PUNTO a Makefile)
- No se utilizan rótulos en clase Robot. (-2 PUNTOS en Uso JavaDoc)
- No se utilizan rótulos en clase RobotView. (-2 PUNTOS en Uso JavaDoc)
- No se agrega diagrama de clases de última etapa desarrollada a archivo de documentación. (-2 PUNTOS en Diagramas de Clases)
- No se realiza una explicación de las clases de la última etapa desarrollada. (-7 PUNTOS en Explicación de Módulos)
- No se mencionan las dificultades encontradas en la tarea ni sus soluciones. (-7 PUNTOS en Dificultades y Soluciones)
- No compila la Etapa 1, por ende, no se puede evaluar. (-15 PUNTOS en Etapa 1)
- No se crea una ventana para redefinir el valor de delta t. (-5 PUNTOS en Etapa 2)
- No se está agregando el robot al mundo, solo se dibuja en el panel. (-5 PUNTOS en Etapa 2)
- Es posible crear un robot sin que haya un maze seleccionado. (-2 PUNTOS en Etapa 2)
- No se actualiza el listado ni la vista de los robots al momento de abrir otro laberinto. (-2 PUNTOS en Etapa 2)
- La vista del robot no se ajusta para que se encuentre dentro del laberinto a utilizar. (-2 PUNTOS en Etapa 2)
- No existe menú para definir velocidad. (-2 PUNTOS en Etapa 3)
- No existe menú para definir piloto. (-2 PUNTOS en Etapa 3)
- Robot puede ser colocado sobre una pared. (-2 PUNTOS en Etapa 3)
- Se puede crear un robot aún estando en simulación (-1 PUNTOS en Etapa 3)
- El robot no se mueve de forma lógica en el laberinto. (-1 PUNTOS en Etapa 3)
- No se presenta Etapa 4, por ende, no se puede evaluar. (-10 PUNTOS en Etapa 4)
- No compila la etapa 1 por falta de archivos fuentes. (-3 PUNTOS en Cero Warning, Cero Errores)

XIX-B. *Recomendaciones*

- Incorporar makefiles para cada etapa.
- Incorporar archivo de documentación.
- No realizar comentarios con muchas tabulaciones y una letra por línea.

XX. GRUPO 19 : 201030013-7 201030003-K

XX-A. *Descuentos*

- No se crea la clase RobotView, por lo que no se considera documentada. (-2 PUNTOS en Uso de JavaDoc)
- Accidentalmente borraron la carpeta **images** que contenía las imágenes de los botones en todos los stages, por lo que estos no se mostraban en la ventana. (No se les descontará por cada uno) (-1 PUNTO en Etapa 1)
- El tamaño de la ventana no se ajusta al tamaño del laberinto, el laberinto es el que se ajusta a la venta, eso puede generar laberintos deformados. (-3 PUNTO en Etapa 1)
- No se actualiza el listado ni la vista de los robots al momento de abrir otro laberinto. (-2 PUNTOS en Etapa 2)
- Robot puede ser colocado sobre una pared. (-2 PUNTOS en Etapa 2)
- Se muestra ventana para entregar valor del delta, pero no realiza cambio alguno en objeto world. (-3 PUNTOS en Etapa 2)
- No se genera ningún archivo que muestre los datos del recorrido de los robots (La muestra del recorrido sólo corresponde a la etapa 5). (-10 PUNTOS en Etapa 4)
- Solo se genera un archivo para el primer robot colocado y además genera un mal formato de salida, por lo que la imagen se ve distorsionada. (-2 PUNTOS en Etapa 5)

XX-B. *Recomendaciones*

- Para esta tarea se pedía el uso del concepto de Modelo-Vista-Controlador, por ello era importante separar el Robot de su vista, así como se separa de su controlador (Pilot), creando una clase RobotView. Esto se hace por varias razones, especialmente por eficiencia y el hecho de poder heredar y crear distintas vistas del robot.
- Cuidado!, la elección de pilotos está al revés, si yo elegía la estrategia por mano derecha utilizaba la izquierda y viceversa.
- Cuidado!, al momento de setear el delta t, en el ejemplo se indica que los decimales se denotan con un punto, pero el código solamente lo acepta si se denotan con una coma.
- Hay que tener cuidado con los inputs, también es necesario limitar las entradas dependiendo de la capacidad del programa, como por ejemplo, un robot con dirección diagonal hará que la lógica no funcione como se espera, además para velocidades muy grandes también pueden haber complicaciones, tenganlo en cuenta!.
- Probar para distintos casos siempre, tratar de romper el código o generar entradas poco convencionales para poder abordar todos los problemas que pudiera llegar a tener el programa y así poder llegar a un código más robusto. Recuerden que los usuarios usualmente son gente sin conocimiento alguno sobre el proyecto o cualquier cosa que tenga que ver con la programación.

XX-C. *Aspectos Destacados*

- Excelente README.
- Muy buena implementación de la etapa 3 y muy buen manejo de las entradas de usuario!.

XXI. GRUPO 20 : 201430040-9 201430035-2 201430052-2

XXI-A. *Descuentos*

- Mensajes de los commits son poco informativos, Muchos commits tienen el mismo mensaje, El nombre del repositorio no sigue el formato solicitado, 201430052-2 realiza menos de 4 commits. Para 201430040-9 y 201430035-2 (-2 PUNTOS a Uso de Repositorio GIT), para 201430052-2 (-6 PUNTOS a Uso de Repositorio GIT)
- No se realiza una explicación de las clases de la última etapa desarrollada. (-7 PUNTOS en Explicación de Módulos)
- No se mencionan las dificultades encontradas en la tarea ni sus soluciones. (-7 PUNTOS en Dificultades y Soluciones)
- El laberinto se muestra rotado con respecto a su imagen original. (-1 PUNTO en Etapa 1)
- No se maneja adecuadamente el caso en que no se ingrese un número en el menú de delta t. (-2 PUNTOS en Etapa 2)
- No se actualiza el listado ni la vista de los robots al momento de abrir otro laberinto. (-2 PUNTOS en Etapa 2)
- La vista del robot no se ajusta para que se encuentre dentro del laberinto a utilizar. (-2 PUNTOS en Etapa 2)
- La vista del robot no incluye a los sensores. (-2 PUNTOS en Etapa 2)
- No se maneja adecuadamente el caso en que no se ingrese un número en el menú de velocidad. (-1 PUNTO en Etapa 3)

- No existe menú para definir piloto. (-2 PUNTOS en Etapa 3)
- No se puede seleccionar la ubicación del robot con el mouse. (-5 PUNTOS en Etapa 3)
- Se implementa funcionamiento para estado de play y para estado de pause, pero no opera correctamente. (-3 PUNTOS en Etapa 3)
- Se puede crear un robot aún estando en simulación (-1 PUNTOS en Etapa 3)
- Los sensores no cambian de color, porque no se visualizan. (-2 PUNTOS en Etapa 3)
- No se presenta Etapa 4, por ende, no se puede evaluar. (-10 PUNTOS en Etapa 4)

XXI-B. Recomendaciones

- No extenderse demasiado en el readme, basta con mencionar como compilar y ejecutar las etapas de la tarea.
- Entregar documentación en formato pdf.
- Basta con agregar el diagrama de alto nivel de la última etapa desarrollada.

XXII. GRUPO 21 : 201604062-5 201630034-1

XXII-A. Descuentos

- No se hace uso de JavaDoc. (-4 PUNTOS en Uso de JavaDoc)
- No se hace uso de GIT. (-8 Puntos en GIT)
- Explicación escueta de la interacción de los módulos. (-3 PUNTOS en Documentación)
- Solo se abordaron 2 Dificultades y una de las soluciones dadas no explica como se resuelve el problema a nivel técnico. (-4 PUNTOS en Documentación)
- El laberinto seleccionado no se almacena en ninguna parte. (-3 PUNTOS en Etapa 1)
- Se deja seleccionar cualquier tipo de archivo sin manejar las excepciones. (-2 PUNTOS en Etapa 1)
- El tamaño de la ventana no se ajusta al tamaño del laberinto. (-3 PUNTO en Etapa 1)
- Los archivos de imagen estaban colocados fuera de la carpeta donde el programa era capaz de reconocerlos, tuvieron que moverse estos archivos para observar el funcionamiento del botón. (-1 PUNTO en Etapa 1)
- El objeto robot no es generado ni almacenado en ningún momento, por lo que tampoco se fijan valores predeterminados para la posición y la velocidad. (-5 PUNTOS en Etapa 2)
- No se manejan excepciones en el set Delta t. (-2 PUNTOS en Etapa 2)
- Se está dibujando un cuadrado en el panel, no se está creando un robot y su representación visual, haciendo además que pueda colocarse sobre las paredes y no tenga ningún tipo de coherencia con el modelo. (-5 PUNTOS en Etapa 2)
- Se muestra ventana para entregar valor del delta, pero no realiza cambio alguno en objeto world. (-3 PUNTOS en Etapa 2)
- No se presenta Etapa 3, por ende, no se puede evaluar. (-20 PUNTOS en Etapa 3)
- No se presenta Etapa 4, por ende, no se puede evaluar. (-10 PUNTOS en Etapa 4)

XXII-B. Recomendaciones

- Se aviso en clase y por correo **reiteradas veces** que al mismo correo les llegaría un mensaje de activación de su cuenta de GIT que los ayudantes hicieron a **principio del semestre**. Luego se volvieron a mandar varios comunicados por correo de que la gente que no alcanzó a activar su cuenta debería mandarles un correo a cualquier ayudante, para que se activaran sus cuentas inmediatamente. La información de la entrega de las tareas debieron haberla leído para la **primera tarea**. La activación de las cuentas de GIT funcionan **todos los días. Hasta el día de hoy a ninguno de los ayudantes nos ha llegado un correo de alguno de ustedes respecto a las cuentas de GIT.**
- Respecto al comentario en la documentación, es su obligación organizar el tiempo para cada uno de sus deberes.
- Si ustedes fallaron en una o más entregas, está bien, pero por favor no den excusas, pidan ayuda a los profesores o ayudantes.

XXIII. GRUPO 22 : 201890108-3 201123008-6 201351025-6

XXIII-A. Descuentos

- Tienen menos del triple del numero de etapas en commits, El nombre del repositorio no sigue el formato solicitado, Realizan commits después del tag Final (en esta ocasión se considerará el último commit, para siguientes, hasta el primer tag final), 201123008-6 y 201351025-6 no realizan commits. Para 201890108-3 (-2 PUNTOS a Uso de Repositorio GIT), para 201123008-6 y 201351025-6 (-8 PUNTOS a Uso de Repositorio GIT)
- No hay makefile. (-1 PUNTO a Makefile)
- No se desarrollan etapas Robot y RobotView como para documentarla. (-4 PUNTOS en Uso JavaDoc)

- No se agrega diagrama de clases de última etapa desarrollada a archivo de documentación. **(-2 PUNTOS en Diagramas de Clases)**
- No se realiza una explicación de las clases de la última etapa desarrollada. **(-7 PUNTOS en Explicación de Módulos)**
- No se mencionan las dificultades encontradas en la tarea ni sus soluciones. **(-7 PUNTOS en Dificultades y Soluciones)**
- No compila la Etapa 1, por ende, no se puede evaluar. **(-15 PUNTOS en Etapa 1)**
- No se presenta Etapa 2, por ende, no se puede evaluar. **(-15 PUNTOS en Etapa 2)**
- No se presenta Etapa 3, por ende, no se puede evaluar. **(-20 PUNTOS en Etapa 3)**
- No se presenta Etapa 4, por ende, no se puede evaluar. **(-10 PUNTOS en Etapa 4)**
- No compila la etapa presentada por falta de archivos fuentes. **(-10 PUNTOS en Cero Warning, Cero Errores)**
- No agregan a los ayudantes a su repositorio. **(3 PUNTOS a Descuentos Varios)**