



PROYECTO DE INGENIERÍA DE SOFTWARE 4

Plan general del proyecto



Estudiantes

Ernesto Lillus Desanti Sciamerelli

Carlos David Madrigal Núñez

Ángel José Marín Trejos

Jose Leonardo Mora Madrigal

Profesores

Álvaro Cordero Peña

Armando Blanco Castro

1er cuatrimestre

26 de febrero del 2020


Documento: Plan general del proyecto	Código: BISOFT-38	
Elaborado por: Mercury	Última modificación: 26 de febrero de 2020	
Revisado por: Ángel Marín Trejos		
Versión: 1.0	Página 2 de 19	

Tabla de contenidos

Introducción.....	4
Plan de administración de la calidad	5
Formato y detalle de las listas de verificación	5
Diseño y especificación de los casos de prueba (pruebas unitarias).....	9
Plan de administración de la configuración.....	11
Descripción de las herramientas.....	11
Administración del repositorio	11
Reglas del repositorio	12
Políticas de respaldo	12
Plan de estándares.....	13
Estándares de interfaz gráfica.....	13
Paleta de colores.....	13
Angular.....	13
Estándares de codificación.....	14
Java.....	14
Angular	14
Android	14
Documentación interna del código.....	15
Javadoc.....	15
Estándares de documentación.....	16
Plan de administración del tiempo	17
Estructura de desglose de trabajo (WBS)	17
Herramientas	19
Lanzamientos y sprints.....	19



Documento: Plan general del proyecto	Código: BISOFT-38 
Elaborado por: Mercury	Última modificación: 26 de febrero de 2020
Revisado por: Ángel Marín Trejos	
Versión: 1.0	Página 3 de 19

Tabla de figuras

Figura 1: Diagrama WBS para el proyecto Ponto. Fuente: Elaboración propia 18


Documento: Plan general del proyecto	Código: BISOFT-38 
Elaborado por: Mercury	Última modificación: 26 de febrero de 2020
Revisado por: Ángel Marín Trejos	
Versión: 1.0	Página 4 de 19

Introducción

El presente documento tiene como objetivo informar acerca de las diversas acciones que se van a realizar para el entendimiento del plan general del proyecto, desde administración correcta de la calidad, una configuración adecuada, los estándares que se deben seguir a lo largo del proyecto y finalizando con el manejo de tiempos, cuyo objetivo es asegurar que todos los aspectos del proyecto se encuentren detallados para que los desarrolladores tengan en cuenta todas las reglas de cada una de estas partes.

Con respecto al plan de calidad se utilizan las listas de verificación, las cuales serán explicadas más adelante, las pruebas unitarias que se realizarán y la manera en la cual se realizarán dichas pruebas siempre adaptado a la metodología Scrum. Para el apartado del plan de configuración se pretende especificar las herramientas que se van a utilizar para el manejo del repositorio, así como la estructura del mismo y las reglas a seguir por cada desarrollador.

En el apartado de estándares se presentan todos los estándares que se utilizarán durante todo el proyecto y las nomenclaturas adecuadas para el código, y en general, los aspectos de estándares de la interfaz de usuario y documentos. Finalmente, en el apartado de tiempo se especifican las herramientas utilizadas para el control del mismo, la descripción del WBS y la estructuración de los lanzamientos y sprints con sus fechas.

Documento: Plan general del proyecto	Código: BISOFT-38 
Elaborado por: Mercury	Última modificación: 26 de febrero de 2020
Revisado por: Ángel Marín Trejos	
Versión: 1.0	Página 5 de 19

Plan de administración de la calidad


Para asegurarse que los documentos cumplan con la calidad respectiva, se utilizarán las denominadas listas de verificación, las cuales evalúan distintos criterios que cumple o no un entregable, ya sea desde el tipo de letra hasta el contenido del mismo, cada documento presenta una lista de verificación propia para un mejor manejo, dichas listas se explican en la sección específica. El subapartado de las pruebas unitarias especifica la herramienta que se pretende utilizar durante el transcurso del proyecto para el manejo de la calidad del producto, así mismo, se especifica en qué momento serán realizadas dichas pruebas.

Formato y detalle de las listas de verificación


Una lista de verificación corresponde al listado de distintos criterios por los cuales se verifica la calidad de los elementos básicos para un documento entregable. Dicha lista se encuentra conformada por una tabla con los siguientes apartados como mínimo: nombre del documento a revisar, descripción del criterio, si el documento cumple o no con dicho criterio y la fecha de la revisión. Con respecto al plazo del proyecto se van a manejar 2 listas de verificación, una por cada documento entregable (Plan general del proyecto y especificación de requerimientos), cuyas listas de verificación se encuentran a continuación.

Lista de verificación para el Plan General del Proyecto

Nombre del documento: Criterio por evaluar	Plan General del Proyecto	
	Estado del criterio	Fecha de revisión
	Sí	No
¿Presenta el documento una tabla de contenidos para mejor navegación y orden?	<input type="checkbox"/>	<input type="checkbox"/>
¿El documento presenta una enumeración coherente y adecuada con respecto a la tabla de contenidos?	<input type="checkbox"/>	<input type="checkbox"/>
¿Presenta el documento una tabla de figuras para cada imagen, tabla o gráfico adjuntado (si aplica)?	<input type="checkbox"/>	<input type="checkbox"/>


Documento: Plan general del proyecto	Código: BISOFT-38	
Elaborado por: Mercury	Última modificación: 26 de febrero de 2020	
Revisado por: Ángel Marín Trejos		
Versión: 1.0	Página 6 de 19	

¿Cada una de las imágenes, tablas y gráficos porta un título y enumeración?	<input type="checkbox"/>	<input type="checkbox"/>
¿Al insertar una imagen, tabla o gráfico se presenta una breve explicación de la misma?	<input type="checkbox"/>	<input type="checkbox"/>
¿Cada párrafo posee la Indentación y espaciado correspondiente?	<input type="checkbox"/>	<input type="checkbox"/>
¿Presenta el documento el mismo tipo de tipografía?	<input type="checkbox"/>	<input type="checkbox"/>
¿Se encuentra el documento libre de ambigüedades?	<input type="checkbox"/>	<input type="checkbox"/>
¿Uso correcto de la plantilla?	<input type="checkbox"/>	<input type="checkbox"/>
¿Se presentan todos los apartados del documento?	<input type="checkbox"/>	<input type="checkbox"/>
¿El documento presenta referencias, citas y bibliografía (si aplica) en el formato APA?	<input type="checkbox"/>	<input type="checkbox"/>
¿Se utiliza el debido encabezado?	<input type="checkbox"/>	<input type="checkbox"/>
¿Se encuentra redactado en tercera persona?	<input type="checkbox"/>	<input type="checkbox"/>
¿El documento no posee errores ortográficos?	<input type="checkbox"/>	<input type="checkbox"/>
¿Documento se encuentra en el formato indicado en un inicio (Microsoft Word)?	<input type="checkbox"/>	<input type="checkbox"/>


Documento: Plan general del proyecto	Código: BISOFT-38 
Elaborado por: Mercury	Última modificación: 26 de febrero de 2020
Revisado por: Ángel Marín Trejos	
Versión: 1.0	Página 7 de 19

Lista de verificación para la Especificación de Requerimientos de Software

Nombre del documento: Criterio por evaluar	Especificación de Requerimientos de Software		Fecha de revisión
	Estado del criterio		
	Sí	No	
¿Presenta el documento una tabla de contenidos para mejor navegación y orden?	<input type="checkbox"/>	<input type="checkbox"/>	
¿El documento presenta una enumeración coherente y adecuada con respecto a la tabla de contenidos?	<input type="checkbox"/>	<input type="checkbox"/>	
¿Presenta el documento una tabla de figuras para cada imagen, tabla o gráfico adjuntado (si aplica)?	<input type="checkbox"/>	<input type="checkbox"/>	
¿Cada una de las imágenes, tablas y gráficos porta un título y enumeración?	<input type="checkbox"/>	<input type="checkbox"/>	
¿Al insertar una imagen, tabla o gráfico se presenta una breve explicación de la misma?	<input type="checkbox"/>	<input type="checkbox"/>	
¿Cada párrafo posee la Indentación y espaciado correspondiente?	<input type="checkbox"/>	<input type="checkbox"/>	
¿Presenta el documento el mismo tipo de tipografía?	<input type="checkbox"/>	<input type="checkbox"/>	
¿Se encuentra el documento libre de ambigüedades?	<input type="checkbox"/>	<input type="checkbox"/>	
¿Uso correcto de la plantilla?	<input type="checkbox"/>	<input type="checkbox"/>	
¿Se presentan todos los apartados del documento?	<input type="checkbox"/>	<input type="checkbox"/>	

Documento: Plan general del proyecto	Código: BISOFT-38 
Elaborado por: Mercury	Última modificación: 26 de febrero de 2020
Revisado por: Ángel Marín Trejos	
Versión: 1.0	Página 8 de 19

¿El documento presenta referencias, citas y bibliografía (si aplica) en el formato APA?	<input type="checkbox"/>	<input type="checkbox"/>
¿Se utiliza el debido encabezado?	<input type="checkbox"/>	<input type="checkbox"/>
¿Se encuentra redactado en tercera persona?	<input type="checkbox"/>	<input type="checkbox"/>
¿El documento no posee errores ortográficos?	<input type="checkbox"/>	<input type="checkbox"/>
¿Documento se encuentra en el formato indicado en un inicio (Microsoft Word)?	<input type="checkbox"/>	<input type="checkbox"/>
¿Los objetivos presentan el formato establecido en la plantilla?	<input type="checkbox"/>	<input type="checkbox"/>
¿Los objetivos se encuentran redactados en infinitivo?	<input type="checkbox"/>	<input type="checkbox"/>
¿Cada uno de los requerimientos posee un identificador único?	<input type="checkbox"/>	<input type="checkbox"/>
¿Cada requerimiento sigue una numeración lógica?	<input type="checkbox"/>	<input type="checkbox"/>
¿Se hace distinción entre los requerimientos funcionales y los no funcionales?	<input type="checkbox"/>	<input type="checkbox"/>

Documento: Plan general del proyecto	Código: BISOFT-38 
Elaborado por: Mercury	Última modificación: 26 de febrero de 2020
Revisado por: Angel Marín Trejos	
Versión: 1.0	Página 9 de 19


Diseño y especificación de los casos de prueba (pruebas unitarias)

Para la realización de las pruebas unitarias de la aplicación se utilizará JUnit debido a que es una herramienta la cual, además de venir incluida en Android Studio, posee un uso de fácil comprensión, lo cual la hace una de las más populares. Cada proceso de la aplicación que presente una importancia elevada deberá tener un caso de prueba para comprobar su estado, así mismo, cada uno de los métodos en las pruebas deben cumplir con los estándares generales del código, es decir, primera palabra en minúscula y cada intercalación entre palabras con letras mayúsculas al inicio de la palabra sin tildes ni caracteres especiales, por ejemplo: metodoDePrueba().

Es importante destacar que se manejará una tabla para las pruebas, la cual es por desarrollador, en donde el nombre del caso de prueba representa un nombre brindado por el programador para identificar la funcionalidad, ejemplo: Iniciar sesión. El identificador representa un código único para la prueba en efecto y este se realiza de la siguiente manera: CP- "Identificador único de la historia de usuario relacionada", ejemplo: CP-HU01. El módulo representa el nivel de funcionalidad superior al cual se asocia la historia de usuario, ejemplo: Sesiones. El método representa la función en la clase de prueba respectiva, ejemplo: inicioSesionPrueba(). El autor representa la persona que programó y probó el caso de prueba. El procedimiento equivale al paso que se debe hacer para realizar la prueba respectiva, para lo cual se adjunta el resultado esperado de la acción y el resultado real que le brinda la aplicación para dicha prueba. Y por último la fecha representa el día/mes/año en que se efectuó la prueba.


<Nombre del caso de prueba>

Identificador			
Módulo			
Método			
Autor			
Fecha			
	Procedimiento	Resultado esperado	Resultado real

Documento: Plan general del proyecto	Código: BISOFT-38 
Elaborado por: Mercury	Última modificación: 26 de febrero de 2020
Revisado por: Ángel Marín Trejos	
Versión: 1.0	Página 10 de 19

Es importante destacar que una historia de usuario no se considera “hecha” hasta que cumpla con la funcionalidad especificada y las pruebas unitarias hayan pasado, así como sus casos de prueba. Se manejarán dos columnas extra en el tablero de actividades, una con el nombre de “Pend. Revisión” y otra con el nombre de “Revisado”, en la primera se posiciona aquella tarea que ya fue completada, pero que no se le ha realizado JUnits ni casos de prueba, mientras que en la segunda se posicionan aquellas tareas que ya se les realizaron los dos puntos solicitados anteriormente, para lo que se adjunta una imagen o un documento mostrando que las pruebas fueron realizadas y ejecutadas correctamente.

Las pruebas son realizadas durante cada sprint por cada uno de los desarrolladores, es decir, se realizan en paralelo con la programación de las piezas de código, si una historia de usuario se encuentra en “Revisado”, pero alguna de sus pruebas no pasó se debe devolver a “En progreso”, arreglar el problema y luego probar de nuevo la historia de usuario (no es necesario probar toda la funcionalidad de la historia de usuario, puede ser del problema como tal, mientras que se adjunte la evidencia que demuestra que ya pasó las pruebas el problema).

Documento: Plan general del proyecto	Código: BISOFT-38 
Elaborado por: Mercury	Última modificación: 26 de febrero de 2020
Revisado por: Angel Marín Trejos	
Versión: 1.0	Página 11 de 19

Plan de administración de la configuración

A continuación, se describe el plan de configuración del Proyecto Ponto, en el cual se especifican las herramientas a utilizar para el manejo de repositorio y respaldo de información, los encargados de la creación de respaldos, la estructura del repositorio y el flujo correcto de este. Así como las reglas a seguir por los miembros del grupo para una correcta administración del repositorio y la estructura que seguir en el almacenamiento de documentación.

Descripción de las herramientas


- Google Drive (Google): Servicio de almacenamiento en la nube, el cual provee 15GB de espacio para cualquier formato de documento lo que permite el compartir los documentos con los miembros del equipo de manera fácil.
- Github (Microsoft): Sistema en la nube de manejo versiones para código que permite compartir proyectos con otras personas, así como la ramificación para efectuar trabajo en paralelo sin afectar el proyecto principal.

Administración del repositorio

Se manejarán 4 branches principales Features, Develop, Release y Master. Además de 1 branch por cada historia de usuario, la cual tendrá su nombre compuesto por el nombre del equipo, nombre del proyecto y código de historia de usuario (Ej. MP-01).

- Features.
- Develop.
- Release.
- Master.

Una vez una historia de usuario se encuentra funcional se agrega a la branch de Features donde la persona que hizo el merge se encargará de verificar la funcionalidad local para luego agregarlo a Develop en el cual se efectuarán las pruebas de Calidad, una vez aprobada por soporte la branch mergeará a Release. El merge con Master solo se efectuará si Release se encuentra funcional o en las fechas límite dadas por los profesores para las presentaciones del proyecto.


Documento: Plan general del proyecto	Código: BISOFT-38	
Elaborado por: Mercury	Última modificación: 26 de febrero de 2020	
Revisado por: Ángel Marín Trejos		
Versión: 1.0	Página 12 de 19	

Reglas del repositorio

- Cada desarrollador debe asegurarse que la branch de su historia de usuario respectiva se encuentre funcional antes de hacer merge con Release.
- Cada desarrollador debe revisar la funcionalidad local de la branch de Release una vez hecho el merge, esto para evitar tener errores locales en los ambientes de despliegue en la nube.
- Solo el rol de soporte posee permisos para hacer merge y push en las branches Release y Máster.
- Los merge a la branch de Release se realizarán una vez a la semana en los viernes a menos de que no haya cambios en la versión de Develop o la versión actual no se pueda desplegar.
- Los commits deben ser diarios en caso de que haya código nuevo, se deberá especificar en el título del commit el número de versión del código subido (Ej. v1) y en los comentarios de este las funcionalidades que se están trabajando y su estado.

Políticas de respaldo

- La documentación se guardará en el drive en el cual cada documento contará con su carpeta específica dentro de la carpeta de documentos.
- Se efectuará un respaldo de la documentación en la cuenta de Google Drive del rol de soporte, así como también se mantendrá una versión local en la computadora de este.
- El respaldo del código se mantendrá de manera local en la computadora del rol de soporte, así como en un repositorio aparte en la herramienta utilizada.
- Los procesos de respaldo se llevarán a cabo los domingos entre 12 p.m. y 12 a.m

Documento: Plan general del proyecto	Código: BISOFT-38	
Elaborado por: Mercury	Última modificación: 26 de febrero de 2020	
Revisado por: Ángel Marín Trejos		
Versión: 1.0	Página 13 de 19	

Plan de estándares

En este subapartado se detallarán los estándares que se utilizarán durante el proyecto, además de la nomenclatura de nombres de archivos, variables, funciones, y carpetas, los cuales el equipo de desarrollo deberá de seguir con el fin de tener un orden al momento de desarrollar el proyecto. También se fijarán los estándares para el desarrollo de la interfaz de usuario, la paleta de colores que se usará en el proyecto y la estructura que llevarán los documentos a entregar.

Estándares de interfaz gráfica

A continuación, se detallarán los estándares a seguir en el desarrollo de la interfaz gráfica del proyecto.


Paleta de colores

En el presente subapartado, se listarán los colores con los que contará la paleta de colores del proyecto en formato hexadecimal:

- #F97304
- #D44206
- #023D67
- #03749C
- #15B1D7
- #5CD4EF

Angular

- El CMS del proyecto deberá ser realizado utilizando Angular Material y sus estándares.

Documento: Plan general del proyecto	Código: BISOFT-38 
Elaborado por: Mercury	Última modificación: 26 de febrero de 2020
Revisado por: Ángel Marín Trejos	
Versión: 1.0	Página 14 de 19

Estándares de codificación

A continuación, se detallarán los estándares a seguir en el desarrollo de la codificación y la documentación del proyecto.

Java


- Los nombres de las variables y los parámetros deberán de estar escritos en el formato *lowerCamelCase*.
- Los nombres de las clases tienen que estar escritos en el formato *UpperCamelCase*.
- Los nombres de los paquetes tienen que estar escritos en el formato *UpperCamelCase*.
- Los nombres de variables y parámetros tienen que ser significativos con base en los datos que contengan.
- Los nombres de los métodos tienen que ser significativos con base en las acciones que ejecuten.

Angular

- Los nombres de los archivos tienen que estar escritos en *snake_case*.

Android

- El proyecto deberá ser realizado utilizando los estandartes de *Material UI*.

Documento: Plan general del proyecto	Código: BISOFT-38 
Elaborado por: Mercury	Última modificación: 26 de febrero de 2020
Revisado por: Ángel Marín Trejos	
Versión: 1.0	Página 15 de 19


Documentación interna del código

A continuación, se detallarán únicamente los elementos que se tienen que documentar y el formato en el que deben de ser documentados.

Javadoc

- Todo método deberá de contar con una pequeña descripción significativa de lo que hace y con la menos las siguientes etiquetas:


Etiqueta	Descripción
@author	Nombre del desarrollador.
@version	Versión del método.
@param	Definición de un parámetro. Requiere una etiqueta por parámetro.
@return	Descripción de lo que el método retorna. No es necesario cuando el método no devuelve nada.
@throws	Definición de las excepciones que el método puede lanzar.

Documento: Plan general del proyecto	Código: BISOFT-38 
Elaborado por: Mercury	Última modificación: 26 de febrero de 2020
Revisado por: Ángel Marín Trejos	
Versión: 1.0	Página 16 de 19

Estándares de documentación

A continuación, se detallarán los estándares que deberán seguir los documentos para poder ser considerados como entregables:

- Los títulos de primer nivel deberán de contar con un tamaño de 14 puntos.
- Los títulos de segundo nivel deberán de contar con un tamaño de 13 puntos.
- Los títulos de tercer nivel deberán de contar con un tamaño de 12 puntos.
- El cuerpo del documento deberá de contar con un tamaño de 11 puntos.
- La fuente que se utilizará para los títulos del documento es Calibri Light.
- La fuente que se utilizará para el cuerpo del documento es Calibri Body.
- El espaciado que se utilizará para el documento es de 1.5 puntos.
- Los párrafos deberán de estar justificados.
- El color para los títulos del documento tiene la siguiente combinación de RGB:
 - Red: 134
 - Green: 78
 - Blue: 168

Documento: Plan general del proyecto	Código: BISOFT-38 
Elaborado por: Mercury	Última modificación: 26 de febrero de 2020
Revisado por: Ángel Marín Trejos	
Versión: 1.0	Página 17 de 19

Plan de administración del tiempo

En el presente subapartado se pretende abarcar todo lo relacionado con la administración del tiempo a lo largo del desarrollo del proyecto. El control y manejo de tiempo utiliza una serie de herramientas que se van a exponer y explicar con mayor profundización a lo largo de este subapartado.

Con el fin de abarcar de la manera más completa la administración del tiempo, se van a utilizar diagramas, tales como el diagrama de Estructura de Desglose del Trabajo (WBS, por sus siglas en inglés), el cual consiste en descomponer, jerárquicamente, el trabajo a ejecutar por el grupo, con el fin de alcanzar los objetivos del proyecto a entregar. Tanto la documentación como los módulos a programar se incluyen y descomponen en dicho diagrama, que será explicado en su sección específica.

Así mismo, se expondrán las diferentes herramientas que se van a utilizar, a lo largo del desarrollo del proyecto, para las diferentes actividades que se deben ejecutar, ya sea escribir las minutas, llevar control de las tareas por hacer, hacer la administración general del proyecto, ver el progreso de cada tarea en los sprints, entre cualquier otra actividad que surja; según lo que dicta la metodología Scrum. Por último, se dará la cantidad de lanzamientos y sprints, así como las fechas para cada uno de los mismos.

Estructura de desglose de trabajo (WBS)

El diagrama de estructura de desglose de trabajo (WBS) muestra las dos secciones principales del proyecto: los documentos y el producto, es decir, la aplicación a desarrollar. En la sección de los documentos se especifican los dos entregables y se desglosan cada uno en los informes o apartados que conforman cada entrega. Por el otro lado, el producto se desglosa en los requerimientos del proyecto, el diseño y los módulos que se pretenden programar a lo largo de las semanas de programación. Para mayor legibilidad del diagrama, se incluyó como un archivo externo titulado *Mercury_WBS.png*.



Documento: Plan general del proyecto	Código: BISOFT-38	
Elaborado por: Mercury	Última modificación: 26 de febrero de 2020	
Revisado por: Ángel Marín Trejos		
Versión: 1.0	Página 18 de 19	



Figura 1: Diagrama WBS para el proyecto Ponto. Fuente: Elaboración propia

Documento: Plan general del proyecto	Código: BISOFT-38 
Elaborado por: Mercury	Última modificación: 26 de febrero de 2020
Revisado por: Ángel Marín Trejos	
Versión: 1.0	Página 19 de 19

Herramientas

A lo largo del desarrollo del proyecto, se utilizarán una serie de herramientas que facilitarán el control del grupo y sus responsabilidades. Para poder llevar un control de las minutas que sea de fácil acceso para todos los integrantes, se utilizará una carpeta compartida, en el drive del equipo, titulada *Minutas*. Como herramienta para el control de tareas se utilizarán, principalmente dos: Trello, para llevar el control de las historias de usuario completadas por cada integrante y Toggl para llevar el control del tiempo que cada integrante les ha dedicado a sus responsabilidades. Por último, se utilizarán aplicaciones como Slack para la comunicación oficial entre los integrantes, drive para compartir documentos, Discord para reuniones grupales de forma virtual y WhatsApp como medio secundario de comunicación.

Lanzamientos y sprints

Según la metodología Scrum, un proyecto debe ser desglosado en lanzamientos, que, a su vez, se deben separar en diversas iteraciones llamadas sprints. El desarrollo de este proyecto va a ser dividido en dos lanzamientos; el primero en el que se pretende entregar el cincuenta por ciento de la solución completa y el segundo el cual se pretende entregar la aplicación totalmente funcional y completa. La entrega del primer lanzamiento se hará el día jueves 26 de marzo del 2020 y la entrega del segundo lanzamiento se hará el día jueves 23 de abril del 2020.

Cada una de estos lanzamientos va a estar compuesto de cuatro sprints, con una duración de una semana cada uno. El primero comenzando el día domingo 1 de marzo del 2020 y concluyendo el domingo de la semana siguiente, es decir, el domingo 8 de marzo del 2020. La asignación del trabajo de cada sprint se hará el día anterior al comienzo del mismo, tomando como base la priorización y estimación de cada historia de usuario, así como los resultados del sprint pasado. Para llevar el control de las tareas y sus estados, los sprints van a ser cargados en la herramienta Trello, con un board por sprint en el cual se crearán las tareas y se van a asignar a los diversos integrantes del equipo de desarrollo.