**DESARROLLO DE UN BOT TELEGRAM PARA AUTOMATIZAR EL REGISTRO DE ASISTENCIA DE ESTUDIANTES UNIVERSITARIOS A CLASES**



**JOSE UBALDO CARVAJAL**

**Estudiante maestría gestión y desarrollo de proyectos de software**

**Profesores**

**Sandra Victoria Hurtado Gil**

**Jorge Iván Meza Martinez**

**UNIVERSIDAD AUTÓNOMA DE MANIZALES**

**MAESTRIA EN GESTIÓN Y DESARROLLO DE PROYECTOS DE SOFTWARE**

**LINEA DE INVESTIGACIÓN EN CALIDAD DE SOFTWARE**

**ÉNFASIS EN DESARROLLO DE SOFTWARE**

**MANIZALES**

**2018**

Contenido

[Contenido 2](#_Toc531243766)

[1. INTRODUCCIÓN, PROPÓSITO DEL TRABAJO 5](#_Toc531243767)

[2. TECNOLOGÍAS/HERRAMIENTAS TÉCNICAS UTILIZADAS 6](#_Toc531243768)

[3. INTRODUCCIÓN A TELEGRAM 7](#_Toc531243769)

[4. DESCRIPCIÓN DEL TRABAJO A REALIZAR 8](#_Toc531243770)

[4.1. Objetivos del trabajo 8](#_Toc531243771)

[4.2. Historia de Usuario “Registro estudiante/Actualización de datos” 8](#_Toc531243772)

[4.2.1. Descripción 8](#_Toc531243773)

[4.2.2. Prototipo no funcional 9](#_Toc531243774)

[4.3. Historia de Usuario “Inscribir asignatura” 10](#_Toc531243775)

[4.3.1. Descripción 10](#_Toc531243776)

[4.3.2. Prototipo no funcional 10](#_Toc531243777)

[4.4. Historia de Usuario “Eliminar inscripción de asignatura” 11](#_Toc531243778)

[4.4.1. Descripción 11](#_Toc531243779)

[4.4.2. Prototipo no funcional 12](#_Toc531243780)

[4.5. Historia de Usuario “Registrar asistencia en asignatura” 12](#_Toc531243781)

[4.5.1. Descripción 12](#_Toc531243782)

[4.5.2. Prototipo no funcional 13](#_Toc531243783)

[4.6. Historia de Usuario “Consultar inasistencias para asignatura” 13](#_Toc531243784)

[4.6.1. Descripción 13](#_Toc531243785)

[4.6.2. Prototipo no funcional 14](#_Toc531243786)

[5. ARQUITECTURA DE LA APLICACIÓN BOT 15](#_Toc531243787)

[6. MODELO DE DOMÍNIO DE APLICACIÓN BOT 16](#_Toc531243788)

[7. LISTA DE REFERENCIA 17](#_Toc531243789)

[8. ANEXOS 18](#_Toc531243790)

**Lista de Figuras**

Figura 1. Bot DeLorean para gestionar recordatorios. Fuente: https://recast.ai/blog/top-telegram-bots-2017/ 7

Figura 2. Prototipo historia de usuario “Registro estudiante” 9

Figura 3. Prototipo historia de usuario “Inscribir asignatura” 10

Figura 4. Prototipo historia de usuario “Inscribir asignatura” 12

Figura 5. Prototipo historia de usuario “Registrar asistencia en asignatura” 13

Figura 6. Prototipo historia de usuario “Consultar inasistencias para asignatura” 14

Figura 7. Arquitectura del *Bot* para registrar asistencia a clases 15

Figura 8. Modelo de domínio del *Bot* para registrar asistencia a clases 16

**Lista de Tablas**

Tabla 1. Tecnologías/herramientas utilizadas. 6

Tabla 2. Historia de usuario “Matricular estudiante”. 9

Tabla 3. Historia de usuario “Inscribir asignatura”. 10

Tabla 4. Historia de usuario “Inscribir asignatura”. 11

Tabla 5. Historia de usuario “Registrar asistencia en asignatura”. 13

Tabla 6. Historia de usuario “Consultar inasistencias para asignatura”. 14

1. INTRODUCCIÓN, PROPÓSITO DEL TRABAJO

En el presente trabajo se pretende desarrollar un Bot para Telegram con objeto de automatizar el registro de asistencia de estudiantes universitarios a clases.

En este documento se registra de forma general el proceso llevado a cabo, así como las actividades necesarias para desarrollar el Bot.

La implementación del proyecto sobre el que se aplicaron los conceptos y herramientas se puede encontrar en el repositorio Github: https://github.com/joseucarvajal/profeJoseBot

1. TECNOLOGÍAS/HERRAMIENTAS TÉCNICAS UTILIZADAS

|  |  |  |
| --- | --- | --- |
| **Tecnología/Herramienta** | **Versión** | **Fuente/Descarga** |
| imagemagick | 8.6 | apt-get install imagemagick |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

Tabla 1. Tecnologías/herramientas utilizadas.

1. INTRODUCCIÓN A TELEGRAM

[Telegram](https://telegram.org/) es una plataforma para mensajería instantánea que incluye una aplicación para los principales sistemas operativos móviles (Android y iOS). Adicionalmente, incluye un cliente escritorio y uno web. Las principales características de Telegram es que las conversaciones son privadas y cifradas, es basado en la nube, es gratis… entre otras (Telegram, 2018). Telegram es uno de los principales competidores de la aplicación [WhatsApp](https://www.whatsapp.com/). Una de las características fundamentales de Telegram, es su capacidad de extensibilidad, en tato permite desarrollar *Bots* que son programas autónomos que pueden interactuar con sistemas o usuarios, de esta manera un usuario de Telegram puede instalar e interactuar con un *Bot* que le presente las noticias. Un ejemplo de aplicación *Bot* es [DeLorean](https://telegram.me/delorean_bot), este *Bot* permite gestionar recordatorios y notificaciones para eventos importantes.

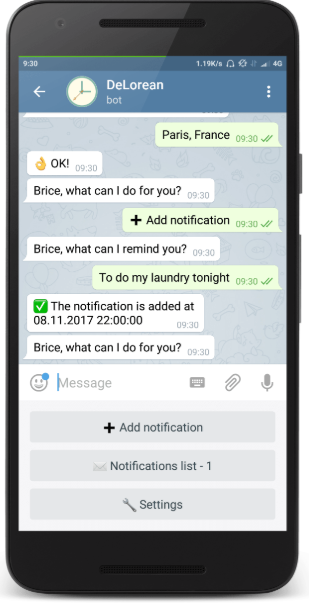


Figura 1. Bot DeLorean para gestionar recordatorios.   
Fuente: <https://recast.ai/blog/top-telegram-bots-2017/>

Los *Bots* se convierten como un contacto más en Telegram de tal manera que los usuarios pueden interactuar con ellos.

1. DESCRIPCIÓN DEL TRABAJO A REALIZAR

En este trabajo se pretende documentar el proceso técnico para la creación y desarrollo de un *Bot* para *Telegram*. Para esto, se construirá un Bot denominado **ProfeJoseBot** que servirá para automatizar el registro de asistencia de estudiantes a clases. Este trabajo se realiza con fines académicos y pretende servir como guía técnica básica a la hora de crear un *Bot* para *Telegram*, por lo cual sus funcionalidades y alcances han sido delimitados y no constituye un proyecto o producto genérico para el sector educativo.

En este apartado se esbozará el alcance del trabajo, las características, casos de uso principales y arquitectura de la aplicación *Bot* para *Telegram*.

* 1. Objetivos del trabajo
* Documentar el proceso para el desarrollo de un *Bot* para *Telegram*.
* Desarrollar un *Bot* que permita registrar la asistencia de estudiantes a clases.

A continuación, se presentan las historias de usuario que se cubrirán en el desarrollo del *Bot* para registro de asistencia a clases:

* 1. Historia de Usuario “Registro estudiante”
     1. Descripción

Permite a un estudiante registrarse en el *Bot*.

|  |  |
| --- | --- |
| **Registro estudiante** | |
| **Descripción** | **Criterios de aceptación** |
| Luego de instalar el Bot en su cuenta de *Telegram*, un estudiante procede a registrarse en el mismo autorizando el envío de su número celular:   1. Número celular: Número celular del estudiante (Requerido). | 1. El estudiante debe aceptar enviar el número de celular para ser almacenado en el sistema *Bot.* 2. El sistema validará que el nro. Celular del estudiante se pertenezca a los estudiantes registrados en el período académico. 3. En caso de que el estudiante no se encuentre registrado, el Bot lo indicará con un mensaje de error. |

Tabla 2. Historia de usuario “Registro estudiante”.

* + 1. Prototipo no funcional

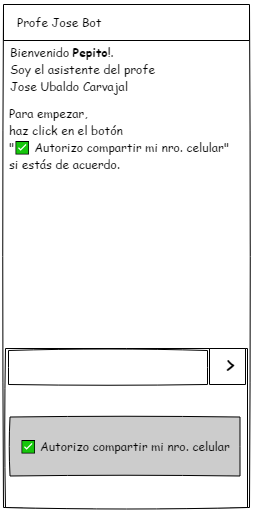


Figura 2. Prototipo historia de usuario “Registro estudiante”

* 1. Historia de Usuario “Visualizar menú principal Estudiante”
     1. Descripción

Una vez el estudiante se ha registrado exitosamente en el Bot, tiene acceso al menú principal.

|  |  |
| --- | --- |
| **Visualizar menú principal Estudiante** | |
| **Descripción** | **Criterios de aceptación** |
| El estudiante visualiza el menú principal con las siguientes opciones:   1. Registrar asistencia. 2. Inscribir asignatura. 3. Des-inscribir asignatura. 4. Editar información básica. | 1. El estudiante puede seleccionar una de las opciones del menú. |

Tabla 3. Historia de usuario “Visualizar menú principal Estudiante”.

* + 1. Prototipo no funcional

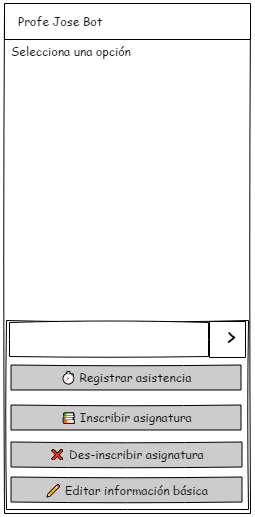


Figura 3. Prototipo historia de usuario “Visualizar menú principal Estudiante”

* 1. Historia de Usuario “Actualizar datos básicos estudiante”
     1. Descripción

Permite a un estudiante editar sus datos básicos en el *Bot*.

|  |  |
| --- | --- |
| **Actualizar datos básicos estudiante** | |
| **Descripción** | **Criterios de aceptación** |
| Luego de instalar el Bot en su cuenta de *Telegram* y registrarse en el mismo, un estudiante procede a actualizar sus datos básicos:   1. Código estudiantil: código del estudiante asignado por la universidad (Requerido). 2. Nombre completo: Nombre completo del estudiante (Requerido). 3. Email: Correo electrónico del estudiante. (Requerido). | 1. El estudiante ingresa los valores requeridos. 2. Una vez el estudiante se ha registrado exitosamente el Bot le indica que ha realizado el proceso satisfactoriamente y se despliega el menú de opciones nuevamente. |

Tabla 4. Historia de usuario “Actualizar datos básicos estudiante”.

* + 1. Prototipo no funcional



Figura 4. Prototipo historia de usuario “Actualizar datos básicos estudiante”

* 1. Historia de Usuario “Inscribir asignatura”
     1. Descripción

Permite a un estudiante inscribirse a una asignatura para reportar asistencia.

|  |  |
| --- | --- |
| **Inscribir asignatura** | |
| **Descripción** | **Criterios de aceptación** |
| Un estudiante ingresa al *Bot* y selecciona la opción “**📅 Registrar asignatura**”. Posteriormente, selecciona una asignatura de un listado y queda asociado a la misma con el fin de registrar asistencia:   1. El estudiante selecciona la opción “**📅 Registrar asignatura**”. 2. El sistema despliega un listado de asignatura correspondientes al periodo actual. 3. El estudiante selecciona una asignatura. | 1. El estudiante queda asociado a la asignatura que seleccionó para registrar asistencia. 2. El sistema informa el resultado de la operación al estudiante. |

Tabla 3. Historia de usuario “Inscribir asignatura”.

* + 1. Prototipo no funcional

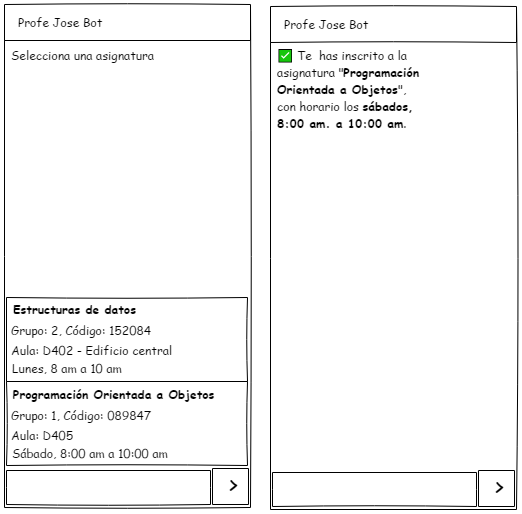


Figura 3. Prototipo historia de usuario “Inscribir asignatura”

* 1. Historia de Usuario “Eliminar inscripción de asignatura”
     1. Descripción

Permite a un estudiante eliminar su inscripción a una asignatura, en el caso que haya seleccionado esta por error.

|  |  |
| --- | --- |
| **Eliminar inscripción de asignatura** | |
| **Descripción** | **Criterios de aceptación** |
| Un estudiante ingresa al *Bot* y selecciona la opción “**❌ Des-inscribir asignatura**”. Posteriormente, selecciona una asignatura de un listado y el sistema elimina la inscripción del estudiante a la misma:   1. El estudiante selecciona la opción “**❌ Des-inscribir asignatura**”. 2. El sistema despliega un listado de asignaturas que tiene inscritas el estudiante para el periodo actual. 3. El estudiante selecciona una asignatura. 4. El sistema elimina la inscripción del estudiante a la asignatura. | 1. El estudiante queda des-asociado a la asignatura que había seleccionado previamente para registrar asistencia. 2. El sistema informa el resultado de la operación al estudiante. |

Tabla 4. Historia de usuario “Inscribir asignatura”.

* + 1. Prototipo no funcional

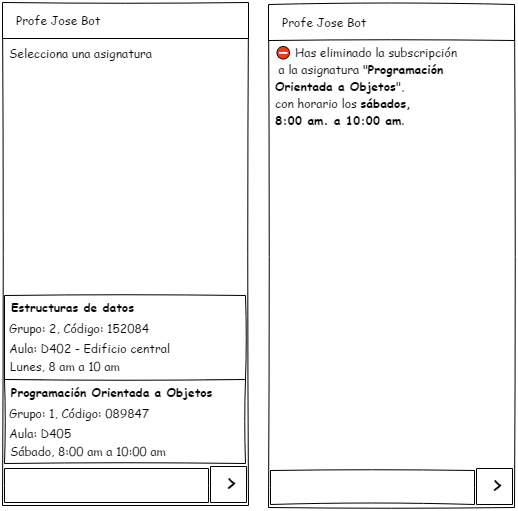


Figura 4. Prototipo historia de usuario “Inscribir asignatura”

* 1. Historia de Usuario “Registrar asistencia en asignatura”
     1. Descripción

Permite a un estudiante registrar la asistencia en una asignatura en particular.

|  |  |
| --- | --- |
| **Registrar asistencia en asignatura** | |
| **Descripción** | **Criterios de aceptación** |
| Un estudiante ingresa al *Bot* y selecciona la opción “**⏱ Registrar asistencia**”. A continuación, el estudiante selecciona una asignatura de un listado y registra asistencia.   1. El estudiante selecciona la opción “**⏱ Registrar asistencia**”. 2. El sistema despliega un listado de asignaturas que tiene registradas el estudiante. 3. El estudiante selecciona una asignatura y el sistema registra la asistencia. | 1. El sistema captura la ***asignatura, fecha, hora y coordenadas espaciales*** donde se encuentra el estudiante y registra estos datos en la asistencia. 2. Si las coordenadas GPS del estudiante no se encuentran dentro del aula de clase, el *Bot* no permite registrar asistencia. 3. Si la hora en la que el estudiante desea registrar asistencia no se encuentra dentro del rango del horario de la asignatura, el *Bot* no permite registrar asistencia. |

Tabla 5. Historia de usuario “Registrar asistencia en asignatura”.

* + 1. Prototipo no funcional

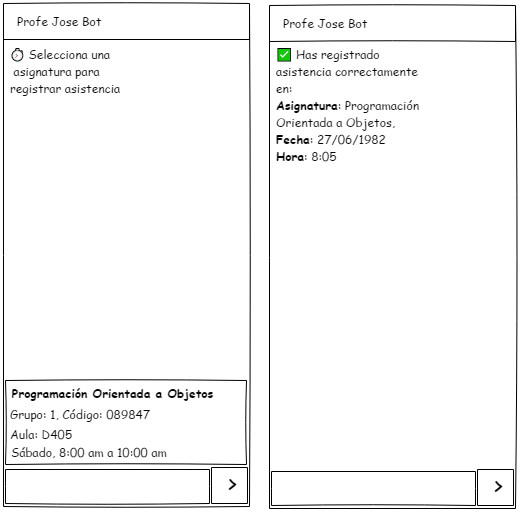


Figura 5. Prototipo historia de usuario “Registrar asistencia en asignatura”

* 1. Historia de Usuario “Consultar inasistencias para asignatura”
     1. Descripción

Le da la posibilidad a un profesor de consultar las inasistencias de una asignatura en particular.

|  |  |
| --- | --- |
| **Consultar inasistencias para asignatura** | |
| **Descripción** | **Criterios de aceptación** |
| El profesor ingresa al *Bot* y selecciona la opción “**📆 Consultar inasistencia**”. A continuación, el docente selecciona una asignatura de un listado y obtiene un reporte de inasistencias para la misma.   1. El docente selecciona la opción “**📆 Consultar inasistencia**”. 2. El sistema despliega un listado de asignaturas que tiene asociadas el profesor. 3. El docente selecciona una asignatura. 4. El sistema despliega los datos de inasistencias registrados para la asignatura. | 1. El sistema valida la identidad del docente según su número celular. 2. Si el número celular del docente no corresponde, el sistema informa de lo sucedido. 3. El sistema despliega las fallas o ausencias registradas con los siguientes datos:    * Fecha (dd/mm/aaaa)    * Código del estudiante    * Nombre del estudiante. |

Tabla 6. Historia de usuario “Consultar inasistencias para asignatura”.

* + 1. Prototipo no funcional

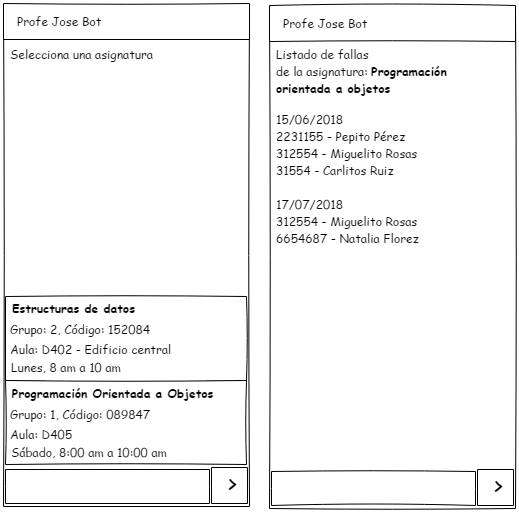


Figura 6. Prototipo historia de usuario “Consultar inasistencias para asignatura”

1. ARQUITECTURA DE LA APLICACIÓN BOT

En este apartado del documento se ilustra la arquitectura de la aplicación *Bot*:

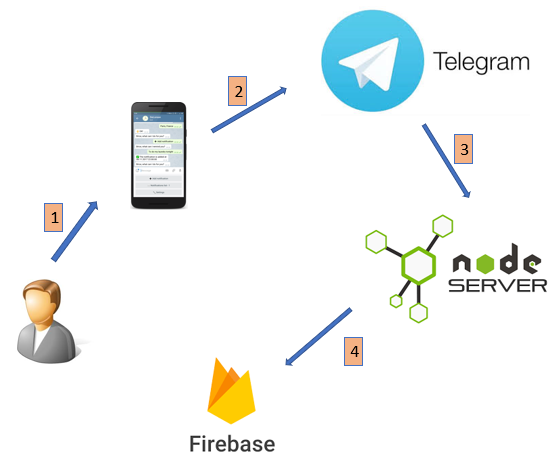


Figura 7. Arquitectura del *Bot* para registrar asistencia a clases

Inicialmente, en el punto 1, un usuario busca e incluye el *Bot* en sus contactos, de esta manera, podrá interactuar con las funcionalidades ofrecidas por el *Bot*,

En el segundo punto, el mensaje es enviado a los servidores de *Telegram*.

En el tercer punto, *Telegram* establece conexión con el servidor *NodeJS*, donde reside la lógica *Back-End* del *Bot*. *Telegram* le envía el mensaje que proviene desde el usuario al servidor *NodeJS*. Adicionalmente, *Telegram* agrega una información adicional al mensaje (chat-id, autor del mensaje… entre otros).

Finalmente, en el paso 4, el servidor *back-end* *NodeJS* establece contacto con el repositorio de datos que en este caso será *Firebase.*

1. MODELO DE DOMÍNIO DE APLICACIÓN BOT

Para la persistencia de datos del *Bot*, se utilizará una base de datos NoSQL de Google denomina *Firebase*. A continuación, se muestra el modelo de dominio de la aplicación:

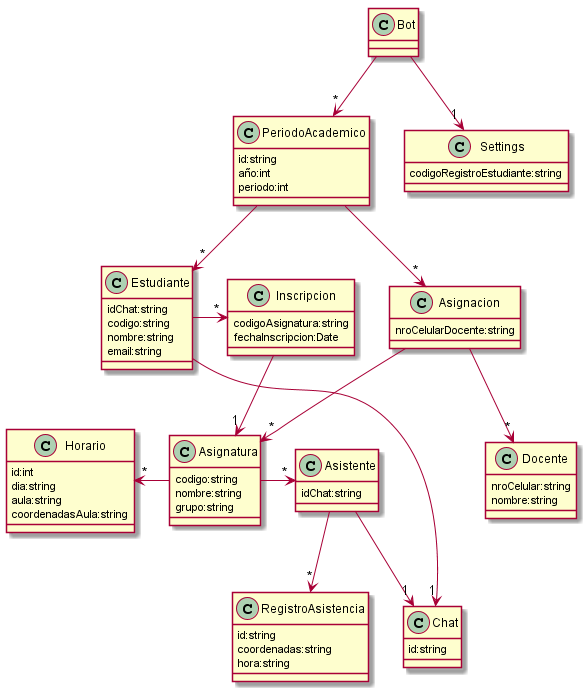


Figura 8. Modelo de domínio del *Bot* para registrar asistencia a clases

1. LISTA DE REFERENCIA

Telegram Official documentation. (2018). Why switch to *Telegram*?. Recuperado de https://telegram.org/.

1. ANEXOS

ANEXO 1. Github de proyecto Laravel matrícula: <https://github.com/joseucarvajal/matricula-laravel>

ANEXO 2. Instancia JSON de ejemplo de la base de datos.