INSTITUTO TECNOLÓGICO AUTÓNOMO DE MÉXICO



Ingeniería de software

Borbotones

Proyecto Final

Tania Ariadna Domínguez Palma

Carlos Armando Ibarrarán Arnaldo

María José Cobacho Trejo

Adrián González Adamuz

Contenidos

[Requerimientos de software 1](#_Toc185634741)

[Plan de Calidad 3](#_Toc185634742)

[Arquitectura y justificación 6](#_Toc185634743)

[Metodología y justificación 8](#_Toc185634744)

[Código del proyecto 9](#_Toc185634745)

[Documentación para replicar 9](#_Toc185634746)

[Propuesta económica 13](#_Toc185634747)

[Video para punto extra 18](#_Toc185634748)

# Requerimientos de software

**Requerimientos funcionales**

* Brindar información general sobre la universidad

Diagrama

Descripción generada automáticamente

* Brindar información sobre actividades y eventos

Diagrama

Descripción generada automáticamente

**Requerimientos no funcionales**

* Accesibilidad: El chatbot debe ser compatible con varios dispositivos móviles.
* Velocidad: Las respuestas deben ser entregadas en menos de 2 segundos para preguntas comunes.
* Seguridad: La información sensible debe ser protegida según estándares de privacidad de datos.

# Plan de Calidad

Este texto describe el plan para el desarrollo del chatbot para guiar a los estudiantes de nuevo ingreso del ITAM, este garantiza que el resultado final cumpla con los requisitos establecidos, proporcionando una experiencia que sea útil, amigable y confiable.

**Objetivos de Calidad**

* El chatbot debe poder responder preguntas sobre inscripción, horarios, ubicaciones en el campus, servicios estudiantiles, etc.
* Las respuestas deben ser precisas, útiles y relevantes.
* El chatbot debe ser fácil de usar, haciendo que las interacciones con los usuarios sean intuitivas.
* El chatbot debe funcionar en cualquier celular que tenga una versión funcional de WhatsApp.
* Cumplir con los estándares de seguridad y privacidad de los datos.
* El sistema debe estar disponible durante las horas estudiantiles.
* Facilitar actualizaciones frecuentes para mejorar el rendimiento y contenido.

**Alcance del plan**

El plan contempla el desarrollo técnico y diseño de las conversaciones, pruebas de usabilidad y el mantenimiento del software.

**Criterios de Calidad**

* Exactitud: Al menos el 80% de las respuestas deben ser correctas.
* Velocidad: El tiempo de respuesta debe ser inferior a 1 segundo.
* Usabilidad: Al menos el 90% de los usuarios deben considerar la interacción intuitiva.
* Compatibilidad: Funcionamiento en cualquier dispositivo con una versión funcional de WhatsApp.
* Seguridad: No se debe presentar ningún caso en donde se filtre la información personal de los usuarios.

**Roles y Responsabilidades**

* Equipo de Desarrollo: Crear el chatbot, implementar las funcionalidades, realizar pruebas técnicas y redactar la documentación.
* Equipo de Pruebas: Evaluar la funcionalidad, usabilidad y compatibilidad.
* Equipo de Mantenimiento: Monitorizar el rendimiento y gestionar actualizaciones.
* Equipo administrativo: Supervisar el cumplimiento de este plan y coordinar las actividades relacionadas.

**Procesos de calidad**

* Planificación: Definir requisitos funcionales y no funcionales.
* Desarrollo: Realizar el chatbot siguiendo buenas prácticas de programación.
* Pruebas: Pruebas unitarias de integración y de sistema, pruebas con usuarios reales para evaluar la experiencia.
* Monitorización: Recopilación de feedback de usuarios.
* Mejora Continua: Incorporar nuevas funcionalidades basadas en el feedback y corregir errores identificados.

**Métricas de Calidad**

* Porcentaje de satisfacción de usuarios (medido a través de encuestas).
* Tasa de resolución exitosa de consultas.
* Frecuencia de errores o fallos.
* Tiempo promedio de actualización y mantenimiento.

**Herramientas y Recursos**

Herramientas de desarrollo: Landbot y github

**Agenda**

***Semana 1: Planificación***

Día 1-3: Establecer alcance, objetivos de calidad, y roles del equipo.

Día 4-5: Redactar y aprobar el plan de calidad.

***Semana 2: Requisitos y Diseño***

Día 6-7: Definir requisitos funcionales y casos de prueba iniciales.

Día 8-9: Diseñar arquitectura técnica y seleccionar herramientas de prueba.

***Semana 3: Desarrollo y Pruebas Iniciales***

Día 10-13: Crear prototipos iniciales y realizar pruebas unitarias.

Día 14: Identificar y corregir errores funcionales básicos.

***Semana 4: Pruebas Funcionales y Usabilidad***

Día 15-16: Pruebas funcionales para validar el flujo de interacción.

Día 17-18: Pruebas de usabilidad con usuarios reales y ajustes.

***Semana 5: Rendimiento y Seguridad***

Día 20-22: Pruebas de carga y auditorías de seguridad.

***Semana 6: Validación y Lanzamiento***

Día 23-25: Validación final y preparación de documentación.

Día 26-28: Aprobación y lanzamiento oficial del chatbot.

Este plan será revisado al final de cada fase del proyecto para asegurar su vigencia y efectividad.

# Arquitectura y justificación

El chatbot está diseñado para manejar flujos de conversaciones predefinidos, por lo que cada interacción del estudiante genera una respuesta predeterminada. Además, está planeado para funcionar en WhatsApp, un método de comunicación instantáneo y bien conocido por la comunidad itamita. Entonces, la arquitectura ideal es una combinación de las siguientes dos.

**Arquitectura basada en eventos**: Esta es la arquitectura principal y está constituida por un disparador, un procesador, y una respuesta a eventos. Un evento se dispara cada vez que un alumno le envía un mensaje al chatbot. El procesador guía la petición por una cadena de respuestas previamente elaboradas (figura 1). La respuesta determinada por el procesador se envía por el chat de regreso para que el estudiante la pueda leer.

A screenshot of a computer screen

Description automatically generated

**Figura 1**. Flujo de respuestas preelaboradas. En rosa se ven las opciones de peticiones que el alumno puede hacer, y en blanco se ven las respuestas que dará el chatbot.

**Arquitectura de microservicios**: Esta arquitectura emplea archivos tipo json para leer y transmitir peticiones a APIs de otras aplicaciones. En este caso, la única API que se usa es la de WhatsApp, a través de la cual se abre el canal de comunicación entre los estudiantes y el chatbot. Sin embargo, queda la arquitectura montada para eventualmente enlazar otros servicios. Un ejemplo podría ser la API de Entérate ITAM, un servicio creado por la oficina de comunicación que se usa para anunciar los eventos institucionales que se llevan a cabo semana con semana.

El flujo total se ve así (figura 2): la interacción sucede a través de la API de WhatsApp, y el resto sucede dentro del bloque de procesadores de eventos que conforman al chatbot.

A diagram of a flowchart

Description automatically generated

**Figura 2**. Arquitectura global de chatbot.

# Metodología y justificación

El producto está dirigido a estudiantes de nuevo ingreso a la universidad – una población de perfiles muy variados y que cambia semestre con semestre. A pesar de que ya son adultos legalmente, la mayoría no cuenta con experiencia previa haciendo trámites ni solicitando servicios académicos. Ocupan orientación clara, accesible, y actualizada, pues las reglas del ITAM también van evolucionando con el tiempo. Por ello, se eligió la siguiente combinación de metodologías, considerando que el desarrollo debe ser altamente adaptable.

**Metodología *feature-driven***: Esta metodología permite descomponer el proyecto en fracciones según las cambiantes necesidades de los estudiantes y las cambiantes exigencias de la institución. Los módulos considerados de primera instancia son: manual sobre bajas de materias, croquis, actividades no académicas disponibles, organizaciones estudiantiles, creación de horarios e información de eventos.

**Metodología por prototipos**: Dentro de los módulos, se planea usar un diseño por prototipos para probar las distintas maneras de interactuar con el chatbot que tendrán los estudiantes. Esta metodología permite crear flujos de conversación variados y validar la eficiencia de cada uno antes de elegir el/los que se usará/n en producción.

**Metodología ágil**: La iteración incremental permite revisitar los módulos para integrar mejoras basadas en retroalimentación y actualizaciones requeridas por la universidad y sus estudiantes. Además, partir los módulos en sprints permite enfocar y aligerar la carga de los desarrolladores.

# Código del proyecto

# Documentación para replicar

**Objetivo**

Implementar un chatbot funcional equivalente al creado en este proyecto.

**Requisitos Previos**

1. Cuenta en [Landbot](https://landbot.io/).
2. Flowchart del chatbot, incluyendo nodos y decisiones.
3. Contenido necesario:
   * Texto del mensaje inicial.
   * Opciones para los botones de la lista.
   * Recursos adicionales, como imágenes o enlaces.

**Pasos para Configurar el Chatbot**

**1. Crear un Nuevo Chatbot**

1. Inicia sesión en [Landbot Dashboard](https://app.landbot.io/).
2. Haz clic en **"Crear Chatbot"** y selecciona el tipo:
   * **WhatsApp**: Para comunicación en WhatsApp.

**2. Configurar el Mensaje Inicial con List Buttons**

1. **Añadir un Bloque de Pregunta:**
   * Arrastra un bloque de **"Botón de lista"** al canvas del editor.
   * Conecta el punto de inicio a este nuevo bloque.
2. **Configurar el Contenido del Mensaje Inicial:**
   * Escribe el texto del mensaje, por ejemplo:
   * Aquí encontrarás toda la información que consideramos imprenscindible para que tengas todo el éxito
   * Añade las opciones según el flowchart:
     + **Opción 1:** "Bajas"
     + **Opción 2:** "Croquis"
     + **Opción 3:** "Actividades en el Itam"
     + **Opción 4:** "Org. Estudiantiles"
     + **Opción 5:** "Armar horario"
     + **Opción 6:** "Eventos"

**3. Diseñar los Flujos Posteriores**

1. **Crear Bloques de Continuación:**
   * De acuerdo al diagrama de flujo, crea y conecta bloques de los siguientes tipos:
     + **"Mensaje"**: Para mostrar respuestas estáticas (texto o imágenes).
     + **"Botón de lista"**: Para capturar información adicional.
2. **Recursos Disponibles:**
   * El diagrama de flujo está disponible en la carpeta FlowDiagram bajo los nombres:
     + Flow1.png
     + Flow2.png
   * Todos los archivos multimedia necesarios para completar el bot están en la carpeta FlowDiagram/Media.

**Instrucciones para Crear Bloques**

**Crear un Bloque de "Mensaje"**

* + Arrastra un bloque de **"Mensaje de texto"** al canvas desde el menú lateral del editor de Landbot.
  + Configura el contenido del mensaje:
  + Escribe el texto que deseas mostrar en el campo de mensaje. Ejemplo:
  + Gracias por tu consulta. Aquí tienes la información que necesitas.
  + (Opcional) Agrega un **archivo multimedia**:
    - Haz clic en el ícono de imagen o adjunta un archivo desde tu computadora.
  1. Conecta este bloque al flujo anterior:
     + Arrastra una línea desde el bloque previo al nuevo bloque de mensaje para garantizar que el flujo continúe correctamente.

**Crear un Bloque de "Botón de lista"**

* + Arrastra un bloque de **"Pregunta"** al canvas desde el menú lateral.
  + Selecciona el tipo de pregunta como **"Botón de lista"**.
  + Configura las opciones:
  + Escribe el texto principal del botón de lista. Ejemplo:
  + ¿Qué categoría te interesa?
  + Agrega opciones personalizadas para el usuario. Ejemplo:
    - Opción 1: "Electrónica"
    - Opción 2: "Ropa"
    - Opción 3: "Hogar"
  1. Conecta cada opción a bloques específicos:
  + Arrastra una línea desde cada opción a los bloques correspondientes que representen las siguientes interacciones en el flujo.

**4. Pruebas**

1. Activa la **vista previa** para probar el flujo.
2. Simula diferentes rutas para verificar que todas las opciones funcionan correctamente.

**5. Publicación**

1. Haz clic en **"Publicar"**.
   * Listo! Tu bot ya está puede recibir mensajes de usuarios.

**6. Optimización Continua**

1. **Seguimiento:**
   * Revisa métricas desde el panel de análisis:
     + Tasa de clics en los botones.
     + Conversaciones completadas.
2. **Ajustes:**
   * Realiza mejoras según los datos y la retroalimentación de los usuarios.

**Notas Finales**

* Monitorea el rendimiento del chatbot regularmente.
* Actualiza el flujo según las necesidades de los usuarios.
* Documenta los cambios importantes para futuras referencias.

# Propuesta económica

Esta sección presenta una propuesta económica detallada para el desarrollo del proyecto, incluyendo la construcción de un menú inicial y el desarrollo de opciones específicas derivadas de este. Además, se considera el impacto y los costos asociados a los roles de apoyo general y un costo fijo relacionado con el equipo. A través de un desglose claro de los días estimados, días reales, capacidad utilizada y costos por rol, esta propuesta busca ofrecer una visión transparente y precisa de la inversión necesaria para llevar a cabo el proyecto con éxito. Todos los cálculos realizados para esta propuesta pueden ser encontrados en el archivo Propuesta Económica/PropuestaEconómica.xlsx.

**Equipo de Trabajo y Costos Individuales**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Rol** | **Salario Mensual** | **Horas Laborales** | **Horas Reales** | **Precio por Hora** | **Precio por Día** |
| Senior Software Developer | $119,000 MXN | 160 | 120 | $992 MXN | $5,950 MXN |
| Junior Software Developer | $22,000 MXN | 160 | 120 | $183 MXN | $1,100 MXN |
| Program Manager Intermedio | $44,000 MXN | 160 | 120 | $367 MXN | $2,200 MXN |
| Diseño UX Intermedio | $23,000 MXN | 160 | 120 | $192 MXN | $1,150 MXN |
| DevOps | $115,000 MXN | 160 | 120 | $958 MXN | $5,750 MXN |
| Diseño UI Intermedio | $23,000 MXN | 160 | 120 | $192 MXN | $1,150 MXN |
| QA | $25,000 MXN | 160 | 120 | $208 MXN | $1,250 MXN |
| **Costo de Equipo** | $50,000 MXN | 160 | 160 | $313 MXN | $1,875 MXN |

**Desglose de costos**

**Menú Inicial**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Funcionalidad | Días Estimados | Días Reales | Rol | Capacidad Utilizada | Costo |
| Menú Inicial | 1 | 1.45 | Diseño UX Intermedio | 25% | $417 MXN |
| Menú Inicial | 1 | 1.45 | Junior Software Developer | 30% | $479 MXN |
| Menú Inicial | 1 | 1.45 | Senior Software Developer | 20% | $1,726 MXN |

**Costo total del menú inicial**: **$2,621 MXN**

**Opción: Bajas**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Funcionalidad | Días Estimados | Días Reales | Rol | Capacidad Utilizada | Costo |
| Bajas | 1 | 1.45 | Diseño UI Intermedio | 20% | $334 MXN |
| Bajas | 1 | 1.45 | Diseño UX Intermedio | 25% | $417 MXN |
| Bajas | 2 | 2.9 | Senior Software Developer | 30% | $5,177 MXN |

**Costo total de la opción "Bajas"**: **$5,927 MXN**

**Opción: Croquis**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Funcionalidad | Días Estimados | Días Reales | Rol | Capacidad Utilizada | Costo |
| Croquis | 2 | 2.9 | Junior Software Developer | 20% | $638 MXN |

**Costo total de la opción "Croquis"**: **$638 MXN**

**Opción: Actividades en el ITAM**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Funcionalidad | Días Estimados | Días Reales | Rol | Capacidad Utilizada | Costo |
| Actividades en el ITAM | 1 | 1.45 | Diseño UI Intermedio | 20% | $334 MXN |
| Actividades en el ITAM | 1 | 1.45 | Diseño UX Intermedio | 15% | $250 MXN |
| Actividades en el ITAM | 2 | 2.9 | Senior Software Developer | 40% | $6,902 MXN |

**Costo total de la opción "Actividades en el ITAM"**: **$7,486 MXN**

**Opción: Org. Estudiantiles**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Funcionalidad | Días Estimados | Días Reales | Rol | Capacidad Utilizada | Costo |
| Org. Estudiantiles | 1 | 1.45 | Diseño UI Intermedio | 20% | $334 MXN |
| Org. Estudiantiles | 1 | 1.45 | Diseño UX Intermedio | 20% | $334 MXN |
| Org. Estudiantiles | 2 | 2.9 | Senior Software Developer | 10% | $1,726 MXN |
| Org. Estudiantiles | 2 | 2.9 | Junior Software Developer | 30% | $957 MXN |

**Costo total de la opción "Org. Estudiantiles"**: **$3,350 MXN**

**Opción: Armar horario**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Funcionalidad | Días Estimados | Días Reales | Rol | Capacidad Utilizada | Costo |
| Armar horario | 1 | 1.45 | Diseño UI Intermedio | 20% | $334 MXN |
| Armar horario | 1 | 1.45 | Diseño UX Intermedio | 25% | $417 MXN |
| Armar horario | 2 | 2.9 | Junior Software Developer | 30% | $957 MXN |
| Armar horario | 2 | 2.9 | Senior Software Developer | 10% | $1,726 MXN |

**Costo total de la opción "Armar horario"**: **$3,433 MXN**

**Opción: Eventos**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Funcionalidad | Días Estimados | Días Reales | Rol | Capacidad Utilizada | Costo |
| Eventos | 1 | 1.45 | Diseño UI Intermedio | 20% | $334 MXN |
| Eventos | 1 | 1.45 | Diseño UX Intermedio | 20% | $334 MXN |
| Eventos | 2 | 2.9 | Junior Software Developer | 30% | $957 MXN |

**Costo total de la opción "Eventos"**: **$1,624 MXN**

**Roles de Apoyo General**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Funcionalidad | Días Estimados | Días Reales | Rol | Capacidad Utilizada | Costo |
| Todas las actividades | 13 | 18.85 | Program Manager Intermedio | 20% | $8,294 MXN |
| Todas las actividades | 13 | 18.85 | QA | 20% | $4,713 MXN |
| Todas las actividades | 13 | 18.85 | DevOps | 20% | $21,678 MXN |

**Costo total roles generales**: **$34,684 MXN**

**Costo Fijo**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Funcionalidad | Días Estimados | Días Reales | Rol | Capacidad Utilizada | Costo |
| Precio Operativo | 13 | 18.85 | - | 23% | $7,952 MXN |

**Costo total fijo**: **$7,952 MXN**

**Resumen y Totalización**

|  |  |
| --- | --- |
| Concepto | Costo |
| Menú inicial | $2,621 MXN |
| Bajas | $5,927 MXN |
| Croquis | $638 MXN |
| Actividades en el ITAM | $7,486 MXN |
| Org. Estudiantiles | $3,350 MXN |
| Armar horario | $3,433 MXN |
| Eventos | $1,624 MXN |
| Roles generales | $34,684 MXN |
| Costo fijo | $7,952 MXN |
| **Subtotal** | $67,714 MXN |
| **IVA (16%)** | $10,834 MXN |
| **Total con IVA** | **$78,548 MXN** |

**Costo total del proyecto**: **$78,548 MXN**

El costo total de desarrollar el Chatbot para alumnos de nuevo ingreso al ITAM es de **$78,548 MXN**. La propuesta presentada ofrece un desglose claro y detallado de los costos asociados al desarrollo del proyecto, garantizando transparencia en cada etapa del proceso. Con un enfoque en la eficiencia y el aprovechamiento óptimo de los recursos del equipo, esta inversión no solo asegura la entrega de un producto funcional y de alta calidad, sino que también establece una base sólida para futuras expansiones o mejoras.

# Video para punto extra

El video es del semestre Otoño 2022. Lo hicimos para Asistencia Preuniversitaria y lo subieron a sus redes como promocional. Curiosamente los cuatro borbotones participamos en él: <https://www.instagram.com/reel/Ciau28YKQRl/?igsh=eDlya2dpOWQ1eWpi>