



**Instituto Politécnico Nacional  
Unidad Profesional Interdisciplinaria de  
Ingeniería campus Zacatecas**

**Área de ubicación para el desarrollo del  
trabajo**

Ingeniería en Sistemas Computacionales

**Línea de investigación**

Cómputo Educativo

Cómputo Móvil

Inteligencia Artificial

**Título del proyecto de Trabajo Terminal**

Diseño y programación de herramienta lúdica  
de la historia de Zacatecas

**Presenta(n):**

Aldo Leija Pardo

Job Adolfo Salinas Hernández

Ramon Emmanuel García Murillo

**Director:**

Dr. Uriel Alejandro Villegas Cuevas

**Asesores:**

Mtro. Efraín Arredondo Morales



Zacatecas, Zacatecas a 26 de febrero de 2024

# Índices

## Índice de contenido



.....	1
Descripción del proyecto.....	2
Objetivo general del proyecto.....	2
Objetivos particulares del proyecto. ....	2
Marco metodológico.....	2
Modelo Cascada .....	2
Cronograma de actividades.....	13
Bibliografía.....	15
<b>Firmas.....</b>	<b>16</b>
<b>Autorización.....</b>	<b>1</b>
<b>Director del proyecto.....</b>	<b>1</b>

## Índice de tablas

## Índice de figuras

## Índice de gráficas

## **Descripción del proyecto.**

El proyecto propuesto consiste en el desarrollo de una aplicación móvil videojuego del tipo “Adivina quién” para el aprendizaje de personas de 12 a 15 años en México sobre la historia de la Toma de Zacatecas.

La aplicación utilizará elementos de juego, como preguntas, pistas y recompensas, para hacer que el aprendizaje de la historia sea más atractivo y divertido para los jóvenes. La aplicación funcionará con modelos de Procesamiento de Lenguaje Natural, los cuales le permitirán al usuario interactuar de forma más agradable. La aplicación generará las pistas que se le proporcionarán al usuario. Además de que mostrará dichas pistas al usuario a través de elementos visuales atractivos.

## **Objetivo general del proyecto.**

Desarrollar una aplicación móvil para la obtención del conocimiento de hechos históricos ocurridos en la Toma de Zacatecas.

## **Objetivos particulares del proyecto.**

- Tener información de fuentes confiables validadas de los hechos históricos por una autoridad de la SEP.
- Contar con un modelo de base de datos de rápido acceso en la nube para las consultas.
- Proporcionar al usuario una interfaz para dispositivos móviles Android, atractiva e interactiva con el usuario, mostrándole pistas del personaje o evento histórico para que el mismo “adivine” la respuesta, o en caso de que éste se rinda, darle el resultado.

## **Marco metodológico.**

### **Modelo Cascada**

El modelo cascada es una metodología de gestión de proyectos secuencial, utilizada principalmente en el desarrollo de software. Se caracteriza por su enfoque lineal, donde cada fase del proyecto se completa antes de pasar a la siguiente [1], [2].

## **Fases del Modelo Cascada**

1. Análisis de requisitos [1], [2].
2. Se definen los objetivos del proyecto y se recopilan las necesidades de los usuarios [1], [2].
3. Diseño: Se crea la arquitectura del software y se diseñan las interfaces de usuario [1], [2].
4. Implementación: Se desarrolla el código fuente del software [1], [2].
5. Pruebas: Se realizan pruebas para detectar y corregir errores [1], [2].
6. Implementación: Se instala el software en el entorno de producción [1], [2].
7. Mantenimiento: Se realizan correcciones de errores y actualizaciones del software [1], [2].

## **Ventajas del Modelo cascada**

- Estructura clara: El proceso es fácil de entender y seguir [1], [2].
- Fácil de planificar: Se puede estimar con precisión el tiempo y el costo del proyecto [1], [2].
- Buena documentación: Se genera una gran cantidad de documentación en cada fase [1], [2].

## **Desventajas del Modelo Cascada:**

- Falta de flexibilidad: Es difícil realizar cambios una vez que se ha iniciado una fase [1], [2].
- Riesgo de errores: Los errores pueden no detectarse hasta las últimas fases del proyecto [1], [2].
- No es adecuado para proyectos con requisitos cambiantes: No se adapta bien a proyectos donde los requisitos no están bien definidos [1], [2].

## **Tipos de proyectos en los que se recomienda usar la metodología cascada**

- 1. Proyectos con requisitos bien definidos [1], [2].
  - ✚ Cuando los requisitos del proyecto son claros, completos y no se esperan cambios importantes [1], [2].
  - ✚ El proyecto tiene un alcance bien definido y no se prevén modificaciones significativas [1], [2].
  - ✚ Los objetivos del proyecto están claramente establecidos y no hay incertidumbre sobre el resultado final [1], [2].
- 2. Bajo riesgo de cambios
  - ✚ Cuando el proyecto tiene un bajo riesgo de cambios en los requisitos o en el entorno del proyecto [1], [2].
  - ✚ El proyecto no está sujeto a cambios externos o a la influencia de factores externos [1], [2].
  - ✚ La tecnología para utilizar en el proyecto es estable y no se esperan cambios importantes [1], [2].

- 3. Proyectos pequeños y simples
  - + Cuando el proyecto es pequeño y simple, con un alcance limitado y un equipo de trabajo reducido [1], [2].
  - + La complejidad del proyecto es baja y no se requieren procesos complejos de gestión [1], [2].
  - + El proyecto tiene una duración corta y no se requieren plazos ajustados [1], [2].
- 4. Presupuesto y tiempo acotados
  - + Cuando el proyecto tiene un presupuesto y un tiempo acotados, y es necesario un enfoque bien definido para controlar los costos [1], [2].
  - + La planificación y el control del proyecto son críticos para el éxito de este [1], [2].
  - + Se requiere una estimación precisa del tiempo y el costo del proyecto [1], [2].
- 5. Entornos regulados
  - + Cuando el proyecto se desarrolla en un entorno regulado que exige un enfoque estructurado y documentado [1], [2].
  - + Se requiere un alto nivel de calidad y cumplimiento de normas en el desarrollo del proyecto [1], [2].
  - + La documentación del proyecto es crítica para la auditoría y el seguimiento de este [1], [2].

## **Modelo Incremental**

El modelo incremental, es una metodología que se trata especialmente de la combinación de los modelos lineal, o bien modelo cascada y prototipo. Consiste en completar varias iteraciones de lo que es el modelo de cascada, pero sin completar ni una. Haciendo iteraciones lo que se hace es crear una evolución en el producto, permitiendo que se agreguen nuevas especificaciones, funcionalidades, opciones, funciones y lo que el usuario requiera después de cada iteración [3] [4].

### **Fases del método incremental**

#### **1. Planificación**

- + Se definen los objetivos del proyecto y se establecen las prioridades [3] [4].
- + Se divide el proyecto en incrementos o entregas [3] [4].
- + Se estima el tiempo y el costo de cada incremento [3] [4].

#### **2. Análisis:**

- + Se recopilan los requisitos del usuario para el primer incremento [3] [4].
- + Se define el alcance del primer incremento [3] [4].

#### **3. Diseño:**

- + Se diseña la arquitectura del software para el primer incremento [3] [4].

- ✚ Se diseñan las interfaces de usuario para el primer incremento [3] [4].

#### 4. Implementación:

- ✚ Se desarrolla el código fuente del software para el primer incremento [3] [4].
- ✚ Se realizan pruebas para detectar y corregir errores en el primer incremento [3] [4].

#### 5. Pruebas:

- ✚ Se realizan pruebas de integración para asegurar que los incrementos se integran correctamente [3] [4].

Se realizan pruebas de aceptación para asegurar que el software cumple con los requisitos del usuario [3] [4].

#### 6. Implementación:

- ✚ Se instala el software en el entorno de producción [3] [4].

#### 7. Mantenimiento:

- ✚ Se realizan correcciones de errores y actualizaciones del software [3] [4].

### **Ventajas del Modelo Incremental**

- Flexibilidad: El modelo incremental es flexible y se adapta bien a cambios en los requisitos [3] [4].
- Reducción del riesgo: El modelo incremental permite detectar y corregir errores en etapas tempranas del proyecto [3] [4].
- Visibilidad del progreso: El modelo incremental permite a los usuarios ver el progreso del proyecto de forma regular [3] [4].
- Mejor control del presupuesto: El modelo incremental permite controlar mejor el presupuesto del proyecto [3] [4].

### **Desventajas del Modelo Incremental**

- Mayor tiempo de desarrollo: El modelo incremental puede tardar más tiempo en completarse que otros modelos de desarrollo [3] [4].
- Mayor complejidad de la gestión: El modelo incremental puede ser más complejo de gestionar que otros modelos de desarrollo [3] [4].
- Mayor riesgo de errores: El modelo incremental puede aumentar el riesgo de errores si no se realiza una buena planificación y gestión [3] [4].

### **El modelo incremental es de gran utilidad en:**

- Proyectos donde no se tienen todos los requerimientos definidos al inicio del proceso [3] [4].
- Equipos de desarrollo medianos o grandes [3] [4].
- Cuando se deben entregar productos en corto tiempo [3] [4].

## Metodología de Prototipos

Se crea una versión simplificada del sistema para comprender mejor las necesidades del usuario. El prototipo se construye rápidamente y se entrega a usuarios, cliente o sus representantes para que lo prueben. Estas personas proporcionan retroalimentación sobre lo que les gusta y no les agrada del prototipo, dicha información se utiliza para actualizar la documentación de los requisitos del sistema y desarrollar el sistema final [4] [5].

### Etapas del método de desarrollo de software de prototipo

El método de desarrollo de software de prototipo se caracteriza por su enfoque iterativo y flexible. Se divide en las siguientes etapas:

1. Definición de objetivos

- ✚ Se establecen los objetivos del prototipo y las necesidades del usuario.
- ✚ Se define el alcance del prototipo [4] [5].

2. Diseño del prototipo

- ✚ Se crea un diseño del prototipo que represente las funcionalidades clave del sistema final [4] [5].
- ✚ Se seleccionan las herramientas y tecnologías para la construcción del prototipo [4] [5].

3. Implementación del prototipo

- ✚ Se construye el prototipo funcional [4] [5].
- ✚ Se realizan pruebas para detectar y corregir errores [4] [5].

4. Evaluación del prototipo

- ✚ Se entrega el prototipo a los usuarios para que lo prueben [4] [5].
- ✚ Se recopila la retroalimentación de los usuarios sobre el prototipo [4] [5].

5. Refinamiento del prototipo

- ✚ Se actualiza el prototipo con base en la retroalimentación de los usuarios [4] [5].
- ✚ Se repiten las etapas 3 a 5 hasta que se obtenga un prototipo satisfactorio [4] [5].

6. Implementación del sistema final

- ✚ Se desarrolla el sistema final utilizando el prototipo como base [4] [5].

### Ventajas del método de desarrollo de software de prototipo

- ✚ Mejora la comprensión de los requisitos del usuario [4] [5].
- ✚ Permite a los usuarios visualizar y experimentar el sistema [4] [5].
- ✚ Identifica y corrige errores en una etapa temprana [4] [5].
- ✚ Reduce el riesgo de desarrollar un sistema que no satisfaga las necesidades del usuario [4] [5].
- ✚ Fomenta la participación del usuario en el proceso de desarrollo [4] [5].

### **Desventajas del método de desarrollo de software de prototipo:**

- ✚ Puede ser más costoso que otros métodos [4] [5].
- ✚ Puede ser difícil crear un prototipo que represente fielmente el sistema final [4] [5].
- ✚ Puede ser difícil obtener una retroalimentación precisa de los usuarios [4] [5].
- ✚ Requiere un equipo con experiencia en el desarrollo de prototipos [4] [5].

### **En qué proyectos se recomienda usar el método de desarrollo de software de prototipo**

- ✚ Proyectos con requisitos poco claros o cambiantes [4] [5].
- ✚ Proyectos donde es importante la participación del usuario [4] [5].
- ✚ Proyectos donde se requiere una rápida validación de las ideas [4] [5].
- ✚ Proyectos con un alto riesgo de errores [4] [5].

### **En qué proyectos suele ser usado el método de desarrollo de software de prototipo**

- ✚ Desarrollo de software web [4] [5].
- ✚ Desarrollo de aplicaciones móviles [4] [5].
- ✚ Desarrollo de sistemas de información [4] [5].
- ✚ Diseño de interfaces de usuario [4] [5].

## **Metodología Espiral**

La metodología en espiral, también conocida como desarrollo en espiral, es un modelo de ciclo de vida del software que combina las características del modelo en cascada y del modelo incremental. Se caracteriza por un enfoque flexible y adaptativo, ideal para proyectos con requisitos cambiantes o donde la gestión de riesgos es crucial [4] [5].

### **Etapas del método de desarrollo de software en espiral**

El método de desarrollo de software en espiral combina las características del modelo en cascada y del modelo incremental. Se divide en las siguientes etapas:

#### **1 Planificación**

- ✚ Se definen los objetivos del ciclo actual y se establecen las prioridades [4] [5].
- ✚ Se analiza el riesgo y se identifican las posibles soluciones [4] [5].
- ✚ Se estima el tiempo y el costo del ciclo actual [4] [5].

#### **2. Análisis**

- ✚ Se recopilan los requisitos del usuario para el ciclo actual [4] [5].
- ✚ Se define el alcance del ciclo actual [4] [5].

#### **3. Diseño**

- ✚ Se crea un diseño del software para el ciclo actual [4] [5].



- ✚ Se seleccionan las herramientas y tecnologías para la implementación del ciclo actual [4] [5].

#### 4. Implementación

- ✚ Se desarrolla el código fuente del software para el ciclo actual [4] [5].
- ✚ Se realizan pruebas para detectar y corregir errores [4] [5].

#### 5. Evaluación

- ✚ Se entrega el software a los usuarios para que lo prueben [4] [5].
- ✚ Se recopila la retroalimentación de los usuarios sobre el software [4] [5].
- ✚ Se revisa el plan del proyecto en base a la evaluación [4] [5].

#### 6. Repetición

- ✚ Se repiten las etapas 1 a 5 hasta que se complete el proyecto [4] [5].

### **Ventajas del método de desarrollo de software en espiral**

- ✚ Manejo del riesgo: Permite identificar y mitigar riesgos de forma temprana [4] [5].
- ✚ Flexibilidad: Se adapta bien a cambios en los requisitos del usuario [4] [5].
- ✚ Control del proyecto: Ofrece un mejor control del tiempo y el costo del proyecto [4] [5].
- ✚ Calidad del software: Promueve un desarrollo de software de alta calidad [4] [5].

### **Desventajas del método de desarrollo de software en espiral**

- ✚ Complejidad: Puede ser más complejo de gestionar que otros métodos [4] [5].
- ✚ Tiempo: Puede llevar más tiempo que otros métodos para completar el proyecto [4] [5].
- ✚ Costo: Puede ser más costoso que otros métodos, especialmente si se requieren varios ciclos [4] [5].

### **En qué proyectos se recomienda usar el método de desarrollo de software en espiral**

- ✚ Proyectos con un alto riesgo de errores [4] [5].
- ✚ Proyectos con requisitos cambiantes [4] [5].
- ✚ Proyectos con un presupuesto y tiempo acotados [4] [5].
- ✚ Proyectos donde se requiere un desarrollo de software de alta calidad [4] [5].

### **En qué proyectos suele ser usado el método de desarrollo de software en espiral**

- ✚ Desarrollo de sistemas de información [4] [5].
- ✚ Desarrollo de software web [4] [5].
- ✚ Desarrollo de aplicaciones móviles [4] [5].
- ✚ Desarrollo de software embebido [4] [5].

## **Metodología Crystal**

Es una familia de metodologías ágiles que se caracteriza por un enfoque flexible, adaptativo y centrado en el ser humano. Se basa en la idea de que el éxito del proyecto depende en gran medida de la colaboración, la comunicación y el empoderamiento del equipo [4] [6].

Esta metodología no tiene un conjunto rígido de reglas o procesos. En cambio, ofrece un marco flexible que se adapta a las necesidades específicas del proyecto [4] [6].

## **Etapas de desarrollo de la metodología**

### **1. Inicio:**

- ✚ Se establece el equipo de proyecto y se definen los roles y responsabilidades [4] [6].
- ✚ Se crea una visión general del proyecto y se establecen los objetivos [4] [6].
- ✚ Se define el ciclo de desarrollo, que típicamente de 1 a 3 semanas [4] [6].

### **2. Planificación:**

- ✚ Se planifican las características del software para el ciclo actual [4] [6].
- ✚ Se estiman el tiempo y el esfuerzo necesarios para cada característica [4] [6].
- ✚ Se definen las tareas y se asignan a los miembros del equipo [4] [6].

### **3. Implementación:**

- ✚ Se desarrolla el código fuente del software para las características del ciclo actual [4] [6].
- ✚ Se realizan pruebas para detectar y corregir errores [4] [6].
- ✚ Se integra el software con los componentes existentes [4] [6].

### **4. Revisión:**

- ✚ Se revisa el software con el equipo y los usuarios [4] [6].
- ✚ Se recopila la retroalimentación sobre el software [4] [6].
- ✚ Se realizan las correcciones y mejoras necesarias [4] [6].

### **5. Implementación:**

- ✚ Se implementa el software en el entorno de producción [4] [6].
- ✚ Se realiza un seguimiento del rendimiento del software [4] [6].
- ✚ Se realizan las correcciones y mejoras necesarias [4] [6].

### **6. Repetición:**

- ✚ Se repiten las etapas 2 a 5 hasta que se complete el proyecto [4] [6].

## **Ventajas del método de desarrollo de software Crystal:**

- ✚ **Flexibilidad** Se adapta bien a cambios en los requisitos del usuario [4] [6].
- ✚ **Enfoque en el usuario:**
- ✚ Promueve la participación del usuario en el proceso de desarrollo [4] [6].
- ✚ **Mejora del trabajo en equipo:** Fomenta la comunicación y la colaboración entre los miembros del equipo [4] [6].
- ✚ **Reducción del riesgo:** Permite identificar y mitigar riesgos de forma temprana [4] [6].

## **Desventajas del método de desarrollo de software Crystal:**

- ✚ **Requiere un equipo experimentado:** El equipo debe tener experiencia en metodologías ágiles y en el desarrollo de software [4] [6].
- ✚ **Puede ser difícil de gestionar:** La planificación y el control del proyecto pueden ser más complejos que en otros métodos [4] [6].
- ✚ **Documentación limitada:** La documentación del proyecto puede ser menos detallada que en otros métodos [4] [6].

## **En qué proyectos se recomienda usar el método de desarrollo de software Crystal:**

- ✚ Proyectos con requisitos cambiantes [4] [6].

- ✚ Proyectos donde se requiere una rápida entrega del software [4] [6].
- ✚ Proyectos con un equipo pequeño y experimentado [4] [6].
- ✚ Proyectos donde se requiere una alta calidad del software [4] [6].

**En qué proyectos suele ser usado el método de desarrollo de software Cristal:**

- ✚ Desarrollo de software web [4] [6].
- ✚ Desarrollo de aplicaciones móviles [4] [6].
- ✚ Desarrollo de sistemas de información [4] [6].
- ✚ Desarrollo de software embebido [4] [6].

## **Metodología Scrum**

La metodología Scrum es un marco de trabajo ágil ampliamente utilizado en el desarrollo de software. Se basa en iteraciones cortas y repetitivas, permitiendo una rápida entrega de valor y la adaptación a los cambios a lo largo del proyecto [4] [5] [6].

### **1. Planificación del Sprint:**

- ✚ Se define el objetivo del Sprint y se seleccionan las funcionalidades que se van a desarrollar [4] [5] [6].
- ✚ Se estima el tiempo y el esfuerzo necesarios para cada funcionalidad [4] [5] [6].
- ✚ Se crea el Sprint Backlog, que es una lista de tareas que se deben completar durante el Sprint [4] [5] [6].

### **2. Sprint:**

- ✚ Se desarrolla el código fuente del software para las funcionalidades del Sprint [4] [5] [6].
- ✚ Se realizan pruebas para detectar y corregir errores [4] [5] [6].
- ✚ Se integra el software con los componentes existentes [4] [5] [6].

### **3. Revisión del Sprint:**

- ✚ Se revisa el software con el equipo y los usuarios [4] [5] [6].
- ✚ Se recopila la retroalimentación sobre el software [4] [5] [6].
- ✚ Se realizan las correcciones y mejoras necesarias [4] [5] [6].

### **4. Retrospectiva del Sprint:**

- ✚ Se analiza el rendimiento del equipo durante el Sprint [4] [5] [6].
- ✚ Se identifican las áreas de mejora [4] [5] [6].
- ✚ Se definen las acciones para mejorar el rendimiento del equipo en el próximo Sprint [4] [5] [6].

### **5. Repetición:**

- ✚ Se repiten las etapas 1 a 4 hasta que se complete el proyecto [4] [5] [6].

### **Ventajas del método de desarrollo de software Scrum:**

- ✚ Flexibilidad: Se adapta bien a cambios en los requisitos del usuario [4] [5] [6].

- ✚ Enfoque en el usuario: Promueve la participación del usuario en el proceso de desarrollo [4] [5] [6].
- ✚ Mejora del trabajo en equipo: Fomenta la comunicación y la colaboración entre los miembros del equipo [4] [5] [6].
- ✚ Reducción del riesgo: Permite identificar y mitigar riesgos de forma temprana [4] [5] [6].
- ✚ Entrega rápida del software: Permite entregar el software en incrementos funcionales de forma regular [4] [5] [6].

#### **Desventajas del método de desarrollo de software Scrum:**

- ✚ Requiere un equipo experimentado: El equipo debe tener experiencia en metodologías ágiles y en el desarrollo de software [4] [5] [6].
- ✚ Puede ser difícil de gestionar: La planificación y el control del proyecto pueden ser más complejos que en otros métodos [4] [5] [6].
- ✚ Documentación limitada: La documentación del proyecto puede ser menos detallada que en otros métodos [4] [5] [6].

#### **En qué proyectos se recomienda usar el método de desarrollo de software Scrum:**

- ✚ Proyectos con requisitos cambiantes [4] [5] [6].
- ✚ Proyectos donde se requiere una rápida entrega del software [4] [5] [6].
- ✚ Proyectos con un equipo pequeño y experimentado [4] [5] [6].
- ✚ Proyectos donde se requiere una alta calidad del software [4] [5] [6].

#### **En qué proyectos suele ser usado el método de desarrollo de software Scrum:**

- ✚ Desarrollo de software web [4] [5] [6].
- ✚ Desarrollo de aplicaciones móviles [4] [5] [6].
- ✚ Desarrollo de sistemas de información [4] [5] [6].
- ✚ Desarrollo de software embebido [4] [5] [6].

A partir de un análisis de las metodologías, se determinó que la metodología en cascada a pesar de ser la más comúnmente seleccionada para Trabajo Terminal en la UPIIZ, existe la desventaja de que en Trabajo Terminal II se cuenta con muy poco tiempo para el desarrollo, además de que requiere de una documentación rigurosamente detallada, así como el resto de las metodologías tradicionales, lo cual es sumamente impráctico considerando que en el análisis contextual para el anteproyecto se determinó que los requerimientos no son muy complejos, además de que por la naturaleza del proyecto, existe la posibilidad de brindarle la opción al cliente de adaptarse al cambio respecto al avance de proyecto.

Por tales motivos se optaría por la metodología incremental, para tener la posibilidad de realizar avances del proyecto en diversas iteraciones (incrementos) dentro de Trabajo Terminal I. Sin embargo, la metodología incremental continúa siendo una metodología tradicional, lo cual también nos obligaría a realizar una planeación detallada y estricta, además de una documentación extremadamente detallada, lo cual no es apto para este proyecto por las razones mencionadas previamente.

**Lo que nos lleva a seleccionar** un framework de la metodología incremental, el cual sería **Scrum**, ya que nos permite realizar los incrementos de una forma mucho más ágil, además de que no requiere de una planeación extremadamente detallada y rigurosa,

sino que se planea de forma general y flexible a cambios conforme va avanzando el proyecto. Además de que nos permite realizar una documentación menos extensa, tanto en el análisis, como en el diseño, lo cual nos permite agilizar el proceso de desarrollo y pruebas.

La gran desventaja de Scrum es que si no existe una persona que no la conozca bien y tenga experiencia en ella, en lugar de beneficiar, puede perjudicar, ya que no se estaría realizando una correcta planeación de trabajo lo que perjudicará al resto de las etapas del ciclo de vida del software. Sin embargo, en este proyecto contaremos con un compañero que ya ha trabajado con esta metodología, lo cual facilitará el proceso de desarrollo. Un integrante del equipo desempeñará el rol de Scrum master y se asegurará de que tanto la planeación, como el resto de las etapas del proyecto se realicen de una manera correcta, de acuerdo con la filosofía ágil y siguiendo los principios de Scrum.

## Cronograma de actividades.

	MAR	ABR	MAYO
Sprints			
▼ <b>TATZ-12</b> Levantamiento de requerimientos <input checked="" type="checkbox"/> TATZ-14 Obtener y clasificar las... TAREAS PO... <input checked="" type="checkbox"/> TATZ-15 Reunión de análisis de... TAREAS PO... <input checked="" type="checkbox"/> TATZ-16 Minuta de la reunión de... TAREAS PO...			
▼ <b>TATZ-13</b> Análisis de riesgos <input checked="" type="checkbox"/> TATZ-17 Reunión de análisis de... TAREAS PO... <input checked="" type="checkbox"/> TATZ-18 Documento de análisis... TAREAS PO... <input checked="" type="checkbox"/> TATZ-19 Minuta de la reunión de... TAREAS PO...			
▼ <b>TATZ-20</b> Chatbot sencillo en Android Studio <input checked="" type="checkbox"/> TATZ-22 Interfaz de chatbot TAREAS PO...			
▼ <b>TATZ-21</b> App que se comunica a una API REST <input checked="" type="checkbox"/> TATZ-23 Interfaz básica de app... TAREAS PO... <input checked="" type="checkbox"/> TATZ-25 Realizar llamadas a un... TAREAS PO... <input checked="" type="checkbox"/> TATZ-26 Mostrar datos obtenido... TAREAS PO...			
▼ <b>TATZ-27</b> API REST básica de prueba <input checked="" type="checkbox"/> TATZ-28 Crear API REST de pru... TAREAS PO... <input checked="" type="checkbox"/> TATZ-29 Conexión de la API con... TAREAS PO... <input checked="" type="checkbox"/> TATZ-52 Construir una base de... TAREAS PO... <input checked="" type="checkbox"/> TATZ-30 Conexión de la API con... TAREAS PO...			
▼ <b>TATZ-31</b> Conectar interfaz a API REST <input checked="" type="checkbox"/> TATZ-32 Añadir elementos visua... TAREAS PO... <input checked="" type="checkbox"/> TATZ-34 Conectar app móvil a l... TAREAS PO... <input checked="" type="checkbox"/> TATZ-35 Realizar petición de "H... TAREAS PO... <input checked="" type="checkbox"/> TATZ-36 Mostrar texto solicitado... TAREAS PO...			
▼ <b>TATZ-37</b> API REST <input checked="" type="checkbox"/> TATZ-33 Crear API REST del pr... TAREAS PO... <input checked="" type="checkbox"/> TATZ-38 Conectar API a un serv... TAREAS PO... <input checked="" type="checkbox"/> TATZ-39 Diseñar base de datos... TAREAS PO... <input checked="" type="checkbox"/> TATZ-40 Construir base de dato... TAREAS PO... <input checked="" type="checkbox"/> TATZ-53 Conectar la API a la ba... TAREAS PO...			
▼ <b>TATZ-42</b> Mostrar pistas en la interfaz del chatbot <input checked="" type="checkbox"/> TATZ-43 Agregar elementos vis... TAREAS PO... <input checked="" type="checkbox"/> TATZ-44 Realizar peticiones a la... TAREAS PO... <input checked="" type="checkbox"/> TATZ-45 Obtener un personaje... TAREAS PO... <input checked="" type="checkbox"/> TATZ-46 Generar 3 pistas del pe... TAREAS PO... <input checked="" type="checkbox"/> TATZ-47 Mostrar las pistas en lo... TAREAS PO...			
▼ <b>TATZ-48</b> La API determina si la respuesta es correc... <input checked="" type="checkbox"/> TATZ-49 La app móvil le envía l... TAREAS PO... <input checked="" type="checkbox"/> TATZ-50 La API procesa la resp... TAREAS PO... <input checked="" type="checkbox"/> TATZ-51 La app móvil muestra e... TAREAS PO...			

**Ver el cronograma completo:**



<https://job-salinas.atlassian.net/jira/software/projects/TATZ/boards/2/timeline?shared=&atlOrigin=eyJpIjojYTUwZjBkYjc4NjNjNDg3N2E2NzgzOTA0OWM2YTk3M2EiLCJwIjoiajI9>

## Bibliografía.

- [1 O. L. Lodoño Palacio, L. F. Maldonado Granados y L. C. Calderón Villafáñez, «Guía para construir Estados del Arte.,» *International Corporation of networks of Knowledge*, p. 39, 2014.
- [2 R. Hernández Sampieri, C. Fernández-Collado y P. Baptista Lucio, *Metodología de la Investigación*, Ciudad de México: Mc. Graw Hill, 2006.
- [3 DIGITAL TALENT AGENCY, «MODELO WATERFALL O EN CASCADA,» [En línea]. Available:  
] [https://www.dtagency.tech/cursos/metodologias\\_gestion\\_proyectos/tema\\_1-ModeloWaterfall.pdf](https://www.dtagency.tech/cursos/metodologias_gestion_proyectos/tema_1-ModeloWaterfall.pdf).
- [4 Instituto Europeo de Posgrado, «Metodología Waterfall: Modelo de gestión de proyectos en cascada,» [En línea]. Available: <https://iep.edu.es/metodologia-waterfall/>.
- [5 UPSIN, «Capítulo II. Antecedentes teóricos,» [En línea]. Available:  
] <http://repositorio.upsin.edu.mx/Fragmentos/Capitulo2cap21952.pdf>.
- [6 UNAM, «Metodologías y procesos de análisis de software,» [En línea]. Available:  
] <http://www.ptolomeo.unam.mx:8080/xmlui/bitstream/handle/132.248.52.100/175/A5%20Cap%C3%ADtulo%202.pdf?sequence=5>.
- [7 FING, «Tecnólogo en informática - Ingeniería de Software,» [En línea]. Available:  
] <https://www.fing.edu.uy/tecnoinf/mvd/cursos/ingsoft/material/teorico/is02b-Tabla%20Comparativa%20Modelos.pdf>.
- [8 REDALYC, «Revisión de metodologías ágiles para el desarrollo de software,» [En línea]. Available:  
] <https://www.redalyc.org/pdf/4962/496250736004.pdf>.



## Firmas.

En esta sección se mostrarán los nombres y las firmas de los alumnos responsables del desarrollo del proyecto de Trabajo Terminal.



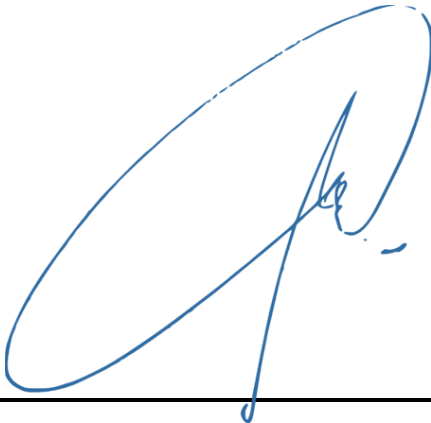
---

Aldo Leija Pardo.  
**Mobile app developer & Product Owner**



---

Ramon Emmanuel García Murillo.  
**Mobile app developer**



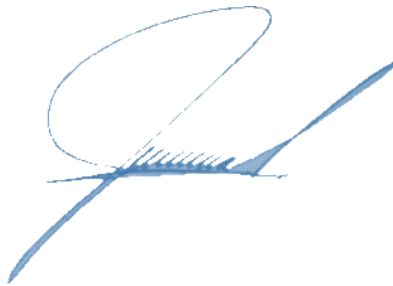
---

Job Adolfo Salinas Hernández.  
**Backend developer & Scrum Master**

## **Autorización.**

Por medio del presente autorizo la impresión y distribución del marco metodológico y cronograma de actividades, toda vez que lo he leído, comprendido en su totalidad, y estar de acuerdo con su desarrollo.

Atentamente;



---

Dr. Uriel Alejandro Villegas Cuevas  
**Director del proyecto**



---

**Asesor**

## Currículum Vitae del director y los asesores del proyecto.

Uriel Alejandro Villegas Cuevas	
	<b>DIRECCIÓN</b> Cerro Azul 314, Colinas del padre, Segunda sección Teléfono: 4921031562 <a href="mailto:urielc@telefonos.com">urielc@telefonos.com</a>
<b>CURP:</b> VICU770610HZSLVR07	

### OBJETIVO

Desarrollarme en trabajos íntegros, responsables y profesionales dentro del ámbito de la comunicación, la investigación y la docencia para colaborar con el desarrollo humano, económico y social de los contextos donde vivo.

### EDUCACIÓN

Licenciado en ciencias de la comunicación

Universidad de LaSalle Bajío.

**Titulado / Cédula profesional 5153149**

Maestro en comunicación y TV. Educativa.

Universidad Autónoma de Durango Campus Zacatecas.

**Titulado / Cédula profesional 7119786**

Maestría en administración.

Universidad Autónoma de Zacatecas.

**Título / Titulación EN TRÁMITE**

Especialista en tecnologías informáticas educativas

Universidad Autónoma de Zacatecas.

**Titulado / Cédula 9244347**

Doctor en ciencias humanísticas y educativas

Universidad Autónoma de Zacatecas.

**Titulado / Cédula 11795415**

## EXPERIENCIA

### **Docente | Instituto Politécnico Nacional (UPIIZ-IPN)**

1 de febrero de 2013 – a la fecha

Docente investigador del área de ciencias sociales y humanidades.

Ha impartido materias de Comunicación, Liderazgo, Administración y Metodología de la investigación.

Reconocido en 2017 como parte de los 20 mejores docentes de la UPIIZ.

Presidente de academia del 2015-2017

### **Profesor y coordinador académico | Universidad Autónoma de Durango Campus Zacatecas**

2002– al 2013

Docente de la licenciatura en comunicación y maestrías como producción audiovisual

Coordinador del sistema Santander y de la preparatoria de la UAD, del 2009-2013.

Se impartió cátedra en preparatoria; las materias de literatura y metodología de la investigación.

En la licenciatura de comunicación se impartieron clases de: Periodismo, Teoría de imagen y sonido (Semiótica); Guionismo, Teorías de la comunicación, Mercadotecnia, Redacción, Análisis del mundo contemporáneo, Seminario de Comunicación y espacio público.

En la maestría de Comunicación audiovisual: Semiótica de la imagen y sonido, Taller de redacción para la producción audiovisual.

### **Profesor | Universidad Autónoma de Zacatecas**

2008– al 2012

Docente suplente de la especialidad en Tecnologías educativas

### **Periodista y locutor | Periódico Imagen y Sonido Estrella**

2000- 2002 y del 2004-2006

Reportero y fotógrafo de fuentes diversas, desde economía hasta sociales, en el periódico IMAGEN

Editor de deportes y editor general de la Fiesta del Balón del Mundial del fútbol Corea-Japón.

Locutor y personificador del programa Tierra Colorada transmitido en Sonido Estrella (Star FM)

## APTITUDES Y HABILIDADES

Investigador de temas de modernidad o historia de comunicación y tecnologías. Ha dirigido tesis a nivel licenciatura y maestría en rubros relacionados con periodismo, semiótica, análisis de contenido, mercadotecnia y redes sociales con resultados notables.

Tengo grandes habilidades de comunicación asertiva y negociación que me han servido como docente y como periodista.

## COMUNICACIÓN

He estado entre los profesores mejor calificados de la UPIIZ desde mi llegada en el 2013.

Tutor académico especializado en resiliencia y acompañamiento educativo.

Facilidad para entender y analizar problemas, además de ejecutar planes para solucionarlos; capacidad de negociación y adaptación, así como de trabajar en equipo y coordinar esfuerzos para la obtención de una meta.

### **LIDERAZGO**

He sido presidente de la academia de ciencias sociales e inglés en la Unidad Profesional Interdisciplinaria de Ingeniería Campus Zacatecas del Instituto Politécnico Nacional (UPIIZ-IPN).

He colaborado con los procesos de reestructuración y reelaboración de las unidades de aprendizaje de comunicación y liderazgo. Además, he encabezado trabajos para el mejoramiento de la impartición de materias referentes a las áreas de sociales y humanidades.

### **AFICIONES**

Soy un ávido lector ecléctico, disfruto de la escritura creativa. Me interesan los deportes de contacto como las artes marciales y la lucha libre; también el baloncesto y el fútbol americano. El campismo, el cine en general.

## Curriculum Vitae

### Datos Personales



Nombre : Efraín Arredondo Morales  
Tel. Particular : (492) 89 9 90 99  
Teléfono Celular : 4921324384  
Correo : [proefra27@gmail.com](mailto:proefra27@gmail.com)

### Preparación Profesional

Maestría : Candidato a Maestría en Tecnologías de la Información  
Lugar : Universidad Interamericana para el Desarrollo (UNID)  
Licenciatura : Ingeniero en Sistemas Computacionales  
Lugar : Instituto Tecnológico Regional de Zacatecas

### Experiencia Laboral

Puesto : **Jefe de la Unidad de Tecnología Educativa y Campus Virtual**  
Lugar : Unidad Profesional Interdisciplinaria de Ingeniería Campus Zacatecas (UPIIZ – IPN)  
Periodo : Enero de 2017 – Septiembre de 2021

Puesto : **Docente**  
Lugar : Unidad Profesional Interdisciplinaria de Ingeniería Campus Zacatecas (UPIIZ – IPN)  
Periodo : Agosto de 2012 – a la Fecha  
Tecnologías Usadas : PHP, MySQL, Unity, Blender, Android, JQuery, SQL, HTML5, JavaScript.

Puesto : **Programador Freelance**  
Periodo : Junio de 2011 – 15 de Julio 2012  
Tecnologías Usadas : PHP, MySQL, Android, javascript (jQuery, jQuery mobile), Ajax, yii framework, CSS3, HTML5, API de facebook y twitter  
Motivo de Separación : Cambio a UPIIZ - IPN

Puesto : **Jefe del Departamento de Informática**  
Lugar : Colegio de Estudios Científicos y Tecnológicos del Estado de Zacatecas (CECyTEZ)  
Periodo : Junio de 2005 – Mayo de 2012  
Motivo de Separación : Cambio de Administración

Actividades Realizadas : **SOPORTE TÉCNICO, CONFIGURACIÓN DE REDES, INTERNET SATELITAL, ENLACES INALAMBRICOS**  
**DESARROLLO DE SISTEMAS DE INFORMACIÓN**  
Tecnologías Usadas: DELPHI, ACCESS, PHP, MySQL, AJAX, API de Google, yiiframework, Librería FPDF y code39  
**IMPLEMENTACIÓN DE MANEJADORES DE CONTENIDO OPEN SOURCE**

Puesto : **Supervisor Escolar**  
Lugar : Dirección General - CECyTEZ  
Periodo : Abril de 2005 – Junio de 2005  
Motivo de Separación : Cambio de Puesto a Jefe del Departamento de Informática

Puesto	: <b>Encargado de Centro de Cómputo</b>
Lugar	: Plantel San José de Lourdes - CECYTEZ
Periodo	: Agosto de 2004 – Abril de 2005
Motivo de Separación	: Cambio de Puesto a Supervisor Escolar
Puesto	: <b>Operador de Sistemas</b>
Lugar	: Programa de Desarrollo Humano Oportunidades
Periodo	: Abril de 2001 –Agosto de 2004
Motivo de Separación	: Cambio a CECYTEZ
Herramientas Usadas	: SQL, Visual FoxPro, Oracle, PL/SQL, PHP
Puesto	: <b>Docente</b>
Lugar	: Instituto Tecnológico Regional de Zacatecas
Periodo	: Enero de 2000 – Agosto de 2002
Motivo de Separación	: Cambio a Oportunidades
Puesto	: <b>Docente</b>
Lugar	: Universidad Autónoma de Zacatecas
Periodo	: Enero de 2000 – Agosto de 2001
Motivo de Separación	: Cambio a Oportunidades
Puesto	: <b>Docente</b>
Lugar	: Instituto Profesional de Computación e Informática S.C.
Periodo	: Agosto de 1999 – Agosto de 2000
Motivo de Separación	: Cambio a Oportunidades



## Plan de trabajo de Trabajo Terminal 1 (TT1)

<b>Project</b>	<b>Diseño y programación de herramienta lúdica de la historia de Zacatecas</b>
<b>Scrum master</b>	Job Adolfo Salinas Hernández.
<b>Product owner</b>	Aldo Leija Pardo.
<b>Approver</b>	<ul style="list-style-type: none"><li>• Uriel Alejandro Villegas Cuevas.</li><li>• Efraín Arredondo Morales.</li><li>• Alfonso Rafael Solís Rangel (Project manager de LABSOL).</li></ul>
<b>Development team</b>	<ul style="list-style-type: none"><li>• Aldo Leija Pardo.</li><li>• Ramón Emmanuel García Murillo.</li><li>• Job Adolfo Salinas Hernández.</li></ul>
<b>Informed</b>	<ul style="list-style-type: none"><li>• Uriel Alejandro Villegas Cuevas.</li><li>• Efraín Arredondo Morales.</li><li>• Julián Rubén Delgado Contreras (Director del LABSOL).</li><li>• Alfonso Rafael Solís Rangel (Project manager del LABSOL).</li></ul>
<b>Objective</b>	Desarrollar una aplicación móvil videojuego que por medio de un chatbot muestre las características de personajes relevantes en la Toma de Zacatecas.
<b>Methodology</b>	Scrum.
<b>Duration</b>	Cinco sprints de dos semanas cada uno.
<b>Due date</b>	Mayo 22, 2024.



## TT1's Scope

### Must have

- El usuario abre la app móvil y la app le muestra tres pistas al usuario en el espacio correspondiente de la interfaz del chatbot.

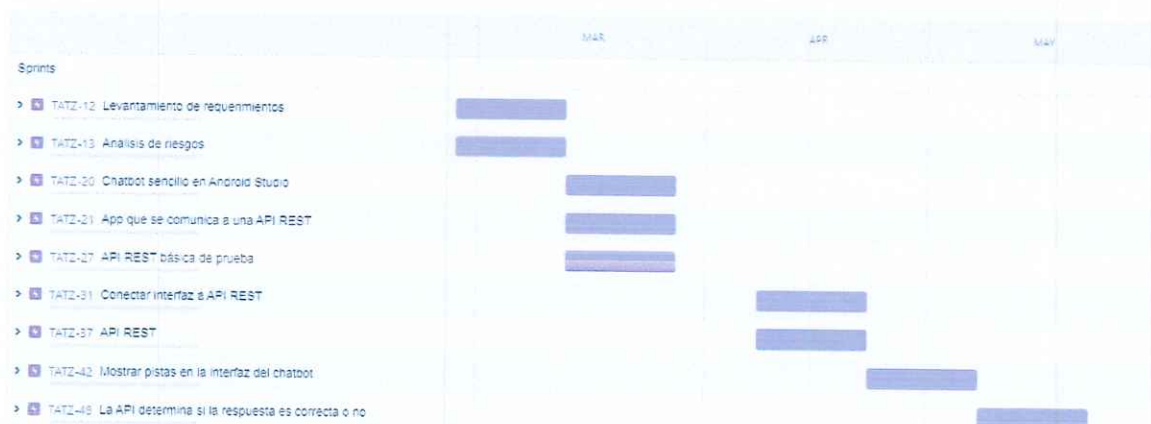
### Nice to have

- El usuario responde y la API determina si la respuesta es correcta o no.
- La app móvil le muestra el resultado al usuario.
- La app móvil primeramente saluda al usuario antes de mostrarle las pistas.
- La app móvil tiene una interfaz atractiva y con elementos visuales llamativos.

### Not in scope

- La app no le proporciona más pistas al usuario.
- La app no le muestra el resultado al usuario.
- La app no le pregunta al usuario si se rinde.
- La app no cuenta con elementos visuales en 3D.
- La app no reproduce o dibuja gráficamente los personajes o eventos históricos.

## Timeline



## Detailed timeline



## Jira timeline

<https://job-salinas.atlassian.net/jira/software/projects/TATZ/boards/2/timeline?shared=&atlassian=eyJpIjoiMjkyYjY1ODJNTg0NGQyYjgyYjY1NDNmNzgxNjc1N2EiLCJwIjoiaU9>

## Backlog

### Sprint 1

Del 27/02/2024 al 12/03/2024.

Objetivo: Realizar el análisis de riesgos y el levantamiento de requerimientos.

Tareas:

- Reunión del equipo de trabajo con el cliente de los posibles riesgos, el impacto de cada uno de ellos y los planes de contingencia.
- Realizar documento de análisis de riesgos.
- Obtener y clasificar las funcionalidades en forma de historias de usuario.
- Reunión del equipo de trabajo con el cliente para analizar las historias de usuario, que este los verifique, se realicen los cambios necesarios en caso de que los haya y valide dichos requerimientos.

### Sprint 2

Del 13/03/2024 al 27/03/2027.

Objetivo: Diseñar y desarrollar interfaz sencilla del chatbot.

Justificación

- El equipo de desarrollo móvil debe ser capaz de obtener datos en formato JSON con peticiones tipo GET de una API REST, desde Kotlin en Android Studio; así como también debe ser capaz de realizar la interfaz de un chatbot.
- El desarrollador backend debe ser capaz de realizar llamadas a un servicio de IA generativa, así como también obtener datos de una base de datos de Firestore.

Tareas

- Realizar llamadas a una API REST para obtener datos desde Android Studio.



Plan de trabajo de TT1 del proyecto: Diseño y programación de herramienta lúdica de la historia de Zacatecas.

- Realizar un chatbot sencillo en Android Studio (realizar interfaz del chatbot).
- Construir una base de datos de prueba en Firestore.
- Realizar una API REST que se conecte a un servicio de IA generativa (como ChatGPT, o Gemini).
- La API REST que se conecte y obtenga datos de la base de datos de Firestore.

### Sprint 3

Del 08/04/2024 al 22/04/2024.

Objetivo: Conectar la interfaz del chatbot a la API REST y a su vez la API se conecte con una IA generativa y con Firestore.

#### Tareas

- Añadir elementos visuales a la interfaz de chatbot.
- Construir API REST del proyecto.
- Conectar la app móvil a la API.
- Realizar petición de "Hola Mundo" a la API.
- Mostrar texto solicitado en el espacio correspondiente del chat.
- Conectar la API a una IA generativa.
- Diseñar base de datos NoSQL.
- Construir base de datos en Firestore.
- Conectar la API a la base de datos de Firestore,

## Sprint 4

Del 23/04/2024 al 07/05/2024.

Objetivo: La app móvil interactúa con la API, a su vez que la API obtiene datos de Firestore y estos datos son procesados en la API para posteriormente ser enviados a la app móvil.

### Tareas

- Agregar elementos visuales adicionales a la interfaz.
- Realizar peticiones a la API para obtener o "pedir" las pistas.
- Obtener un personaje o evento histórico aleatorio de la base de datos desde la API REST.
- Generar 3 pistas del personaje seleccionado.
- Mostrar las pistas en los espacios correspondientes del chat.

## Sprint 5

Del 08/05/2024 al 22/05/2024.

Objetivo: La API recibe la respuesta del usuario y determina si es correcta o no y muestra el resultado.

### Tareas

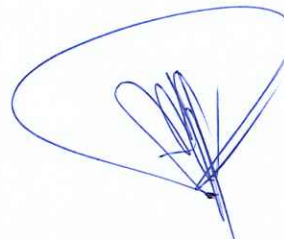
- La app móvil le envía la respuesta del usuario a la API.
- La API auxiliándose de la IA generativa procesa la respuesta a partir de los datos del personaje y determina si la respuesta es correcta o no.
- Le envía la respuesta a la app móvil.
- La app móvil muestra la respuesta al usuario en el espacio correspondiente del chatbot.

Plan de trabajo de TT1 del proyecto: Diseño y programación de herramienta lúdica de la historia de Zacatecas.



---

Dr. Uriel Alejandro Villegas Cuevas  
**Director del proyecto**



---

Mtro. Efraín Arredondo Morales  
**Asesor**



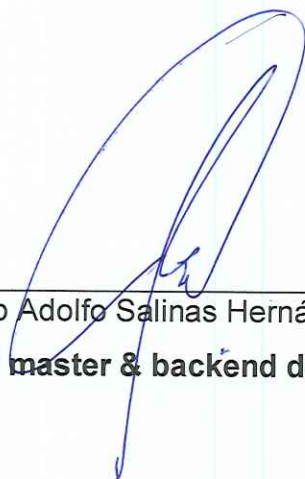
---

Aldo Leija Pardo  
**Product owner & mobile app  
developer**



---

Ramón Emmanuel García Murillo  
**Mobile app developer**



---

Job Adolfo Salinas Hernández  
**Scrum master & backend developer**