



UNIVERSIDAD DE GRANADA

Prácticas 1-2

Planificación y Gestión de Proyectos Informáticos

Máster Universitario en Ingeniería Informática

Autores

- Pablo Morenilla Pinos
- Miguel Morocho Ramos
- Pablo Valenzuela Álvarez

Índice

Propuesta técnica	3
Título: AppRendo Solo	3
Resumen	3
Lugar de ejecución	3
Objetivos: objetivo general y objetivos específicos	3
Justificación	4
Antecedentes	4
Innovación	4
Lista de actividades a realizar alineadas con los objetivos	4
Cronograma*: actividades agrupadas en paquetes de trabajo organizadas en el tiempo, implicación de los participantes y recursos asignados. Entregables para cada actividad	7
Cauces de seguimiento por parte del cliente	8
Garantía o mantenimiento	8
Valor añadido	8
Manual de coordinación o plan de dirección	8
METODOLOGÍA Y CICLO DE VIDA:	8
RECURSOS HARDWARE DESARROLLO:	9
RECURSOS SOFTWARE DESARROLLO:	10
Organización del Equipo de Trabajo (Estructura, Normas):	11
Herramientas para Comunicaciones en el Equipo de Trabajo:	11
Relaciones con el Cliente (Entrevistas, Reuniones, Revisiones):	11
Estándares de Documentación:	11
Estándares de Código:	11
Medición del Desempeño o Progreso:	12
Plan de Gestión de Cambios:	12
Control de Versiones (Método y Herramientas):	12
Gestión de Calidad (Proceso y Herramientas):	12

Propuesta técnica

Título: AppRendo Solo

Resumen

Hay que realizar una aplicación para el Colegio de Educación Especial Clínica San Rafael. Este colegio es un centro concertado que atiende a alumnos con diversidad funcional con edades entre los 3 y los 21 años.

El proyecto está centrado en ayudar a los alumnos a obtener más autonomía mediante su integración en el mundo laboral, por lo que el proyecto debe cumplir ciertas pautas:

- La aplicación ha de ser sencilla de usar, preferiblemente móvil.
- Debe permitir al centro organizar y gestionar tareas para cada alumno y saber si las realiza mediante la supervisión de los administradores/educadores.
- Marcar el feedback de las tareas realizadas por parte del alumnado.
- Habilitar una comunicación entre los educadores que supervisan una tarea mediante un chat.
- Debe tener la opción de crear gráficas relacionadas con las actividades realizadas por el alumnado, estas gráficas deben ser sencillas porque tienen que ser comprensibles por los alumnos.
- En la gestión de las tareas hay que añadir una monitorización del alumno, para saber en el lugar en el que se encuentran durante la realización de las tareas y con quién (dentro de una lista de personas registradas). Se supone que dentro del horario escolar.
- Por último, debe analizar los datos provenientes de un reloj inteligente, que medirá el estado emocional del alumno mientras realiza las tareas de su agenda.

Lugar de ejecución

Oficina del equipo de desarrollo, edificio del CETIC.

Objetivos: objetivo general y objetivos específicos

1. Entregar un Producto Funcional: El objetivo principal suele ser desarrollar y entregar un producto informático que cumpla con los requisitos y expectativas del cliente.
2. Cumplir con Plazos y Presupuesto: Establecer fechas límite realistas y gestionar eficientemente los recursos para cumplir con el cronograma y el presupuesto del proyecto.

3. Calidad del Producto: Garantizar que el producto final sea de alta calidad, sin errores significativos y que cumpla con los estándares de la industria.
4. Usabilidad: Desarrollar una interfaz de usuario intuitiva y fácil de usar, asegurando que el producto sea accesible para su público objetivo.
5. Integración con Sistemas Existentes: Si es necesario, asegurar la integración efectiva del nuevo sistema con sistemas existentes en la organización.
6. Documentación: Crear documentación completa y comprensible que describa la arquitectura, el funcionamiento y el mantenimiento del sistema.

Justificación

En el mercado hay varias aplicaciones destinadas a ayudar a gente con diversidad funcional:

1. Co:Writer Universal:
 - a. Objetivo: Ayudar a estudiantes con dificultades de escritura y ortografía.
 - b. Características: Ofrece predicción de palabras en tiempo real y funciones de corrección ortográfica para facilitar la escritura.
2. Kahoot!:
 - a. Objetivo: Involucrar a estudiantes con discapacidades en actividades educativas interactivas.
 - b. Características: Permite a los educadores crear cuestionarios y juegos interactivos que pueden adaptarse para satisfacer las necesidades de los estudiantes con diversas habilidades.
3. Visual schedules - Autism (AutiPlan):
 - a. Objetivo: Ayudar a estudiantes con trastorno del espectro autista (TEA) a seguir rutinas y comprender actividades.
 - b. Características: Permite la creación de horarios visuales personalizados para guiar a los estudiantes a través de las actividades diarias.

Antecedentes

Somos una Pyme con varios años de experiencia, hemos realizado contratos relacionados con la Universidad de Granada a modo de soporte, testeo y control de calidad.

Innovación

Los proyectos anteriores iban relacionados con controles de calidad, este proyecto supone algo nuevo para nuestro equipo.

Lista de actividades a realizar alineadas con los objetivos

Objetivo General:

Entregar un Producto Funcional

Definición de Requisitos:

- Reunión con administradores y educadores para identificar requisitos específicos de la aplicación.
- Documentación detallada de los requisitos y expectativas del cliente.
- Diseño de Interfaz de Usuario:
- Creación de prototipos de interfaz de usuario sencillos y móviles.
- Validación de prototipos con usuarios potenciales (administradores, educadores y estudiantes).

Desarrollo de la Aplicación:

- Implementación de la aplicación siguiendo las mejores prácticas de desarrollo de software.
- Integración de funcionalidades de gestión de tareas, chat, historial semanal y monitoreo de ubicación.

Pruebas y Validación:

- Realización de pruebas unitarias y de integración para garantizar la calidad del código.
- Sesiones de prueba con usuarios finales para evaluar la usabilidad y la eficacia de la aplicación.

Despliegue y Entrega:

- Configuración de un entorno de producción.
- Entrega de la aplicación funcional al Colegio de Educación Especial Clínica San Rafael.

Objetivo Específico:

- Cumplir con Plazos y Presupuesto

Planificación del Proyecto:

- Estimación de tiempos y recursos para cada fase del proyecto.
- Creación de un cronograma detallado con fechas límite realistas.

Seguimiento del Progreso:

- Implementación de un sistema de seguimiento para controlar el avance del proyecto.
- Identificación y manejo proactivo de posibles desviaciones en el cronograma.

Pruebas Rigurosas:

- Realización de pruebas exhaustivas para detectar y corregir posibles errores.
- Garantizar que la aplicación cumple con estándares de calidad predefinidos.

Retroalimentación Continua:

- Establecimiento de un canal de retroalimentación con usuarios finales.
- Implementación de mejoras basadas en la retroalimentación recibida.
- Usabilidad

Diseño Centrado en el Usuario:

- Incorporación de comentarios de usuarios en la fase de diseño.
- Iteraciones en el diseño basadas en pruebas de usabilidad.

Capacitación y Soporte:

- Creación de materiales de capacitación para administradores, educadores y estudiantes.
- Ofrecimiento de soporte técnico para resolver posibles problemas de usabilidad.

Objetivo Específico:

- Integración con Sistemas Existentes

Análisis de Requisitos de Integración:

- Identificación de sistemas existentes en el colegio que deben integrarse.
- Desarrollo de soluciones para la integración efectiva.
- Documentación

Documentación Completa:

- Creación de documentos detallados que describen la arquitectura y funcionamiento de la aplicación.
- Manuales de usuario y mantenimiento para garantizar una comprensión completa del sistema.

Cronograma*: actividades agrupadas en paquetes de trabajo organizadas en el tiempo, implicación de los participantes y recursos asignados. Entregables para cada actividad

Fase del Proyecto	Inicio	Fin	Duración (en días laborables)	Responsable
Definición de Requisitos	03/04/2023	14/04/2023	10 días	Todos los equipos
Diseño de Interfaces	17/04/2023	28/04/2023	10 días	Equipo de diseño
Desarrollo de la Aplicación	03/05/2023	23/08/2023	80 días	Equipo de desarrollo
Pruebas y Validación	21/08/2023	01/09/2023	10 días	Equipo de pruebas
Capacitación y Soporte	04/09/2023	07/09/2023	4 días	Todos los equipos
Despliegue y Entrega	04/09/2023	15/09/2023	10 días	Todos los equipos

Cauces de seguimiento por parte del cliente

El seguimiento por parte del cliente se realizará mediante reuniones periódicas y realización de test.

Garantía o mantenimiento

Para la gestión de incidencias se seguirá la siguiente lista:

- Identificación, registro y clasificación de incidencias.
- Asignación de responsabilidades.
- Investigación y desarrollo de soluciones.
- Test y validación de incidencias.
- Cierre de incidencia.

Valor añadido

El desarrollo de esta aplicación puede generar un considerable valor añadido, tanto desde una perspectiva social como comercial. Estos son algunos de los aspectos que resaltan el valor de este tipo de aplicación.

1. Facilitación del proceso educativo.
2. Apoyo a educadores y personal escolar.
3. Mejora de la imagen de nuestra empresa.
4. Adaptabilidad y escalabilidad de nuestra aplicación en caso de que tenga éxito.
5. Feedback positivo por parte del usuarios.

Manual de coordinación o plan de dirección

METODOLOGÍA Y CICLO DE VIDA:

Para este proyecto se usará una metodología ágil, como Scrum

1. Recolección de requisitos:

- Reuniones con el equipo del centro educativo para entender sus necesidades.
- Definición clara de las características y funciones clave de la aplicación.

2. Planificación:

- Desglose de las funciones en historias de usuario y definición de criterios de aceptación.
- Asignación de roles y responsabilidades en el equipo de desarrollo.

3. Desarrollo iterativo:

- Creación de versiones incrementales de la aplicación con características específicas.
- Revisiones regulares con el equipo del centro educativo para obtener retroalimentación.

4. Pruebas:

- Pruebas unitarias y de integración después de cada iteración.
- Pruebas de aceptación con el equipo del centro educativo para garantizar que cumple con sus expectativas.

5. Implementación:

- Lanzamiento gradual de la aplicación, comenzando con un grupo piloto.
- Monitoreo de la aplicación para abordar problemas inesperados.

6. Evaluación continua:

- Recopilación de comentarios de usuarios y ajuste continuo según sea necesario.
- Implementación de mejoras y nuevas funciones en ciclos iterativos.

7. Mantenimiento:

- Resolución de problemas y corrección de errores.
- Actualizaciones periódicas para garantizar la compatibilidad con nuevos dispositivos o sistemas operativos.

RECURSOS HARDWARE DESARROLLO:

Para desarrollar la aplicación como la descrita, necesitaremos varios recursos hardware:

1. Computadoras:
 - Estaciones de trabajo para desarrolladores con potentes capacidades de procesamiento.
2. Dispositivos móviles:
 - Smartphones y tablets para probar la aplicación en diferentes plataformas y tamaños de pantalla.
3. Servidores:
 - Servidores para alojar la aplicación y la base de datos, ya sea en la nube o localmente.
4. Relojes inteligentes:
 - Dispositivos de prueba que admitan la integración con sensores para recopilar datos emocionales.
5. Dispositivos de grabación:
 - Cámaras y micrófonos para grabar videos y audios que los estudiantes puedan subir como evidencia de tareas realizadas.
6. Sensores para reloj inteligente:
 - Sensores de ritmo cardíaco u otros dispositivos que puedan medir el estado emocional.
7. Equipo de red:
 - Conexión a Internet de alta velocidad para facilitar el desarrollo y las pruebas en tiempo real.
8. Dispositivos de localización:
 - En caso de querer implementar la monitorización de la ubicación, podrías necesitar hardware que admita tecnologías de geolocalización.
9. Dispositivos para pruebas de hardware:
 - Equipos para probar la integración de la aplicación con diferentes dispositivos y asegurar la compatibilidad.

Asegurarte de tener hardware adecuado para el desarrollo y pruebas es esencial para garantizar un producto final robusto y funcional. ¿Algo más en lo que pueda ayudarte?

RECURSOS SOFTWARE DESARROLLO:

1. Entorno de Desarrollo Integrado (IDE):
 - Puedes utilizar IDEs como Visual Studio Code, IntelliJ, o Android Studio, dependiendo de las plataformas objetivo.
2. Lenguajes de Programación:
 - Para una aplicación móvil, podrías utilizar Java o Kotlin para Android y Swift o Objective-C para iOS.

3. Framework de Desarrollo:

- Para facilitar el desarrollo, considera el uso de frameworks como React Native, Flutter o Xamarin, que permiten construir aplicaciones móviles multiplataforma.

4. Sistema de Gestión de Versiones:

- Git es esencial para el control de versiones. Plataformas como GitHub o GitLab pueden ser utilizadas para alojar repositorios.

5. Base de Datos:

- Utiliza bases de datos como MySQL, PostgreSQL o MongoDB para almacenar información sobre tareas, usuarios y otras entidades relevantes.

6. Servidor Web:

- Puedes necesitar un servidor web para manejar la lógica de negocio y las interacciones con la base de datos. Node.js, Django, o Flask son opciones comunes.

7. Plataforma de Desarrollo Móvil:

- Para desarrollo Android, el kit de desarrollo de software (SDK) de Android. Para iOS, el Xcode con el iOS SDK.

8. Herramientas de Pruebas:

- Frameworks como JUnit para pruebas unitarias, y herramientas específicas para pruebas de aplicaciones móviles como Appium o Espresso.

9. Herramientas de Colaboración:

- Plataformas como Slack, Trello o Asana para facilitar la comunicación y la gestión de tareas dentro del equipo.

10. Herramientas de Diseño:

- Adobe XD, Sketch, o Figma para diseñar y prototipar la interfaz de usuario.

11. Plataforma de Nube:

- Servicios en la nube como AWS, Azure o Google Cloud para alojar la aplicación y la base de datos.

12. Herramientas de Monitoreo:

- Herramientas como Firebase Analytics o Google Analytics para rastrear el uso de la aplicación y obtener datos de análisis.

Organización del Equipo de Trabajo (Estructura, Normas):

- La estructura organizativa del equipo se basará en un enfoque colaborativo, con roles claramente definidos para desarrolladores, diseñadores y personal de pruebas. Se establecerán normas de comunicación efectiva y reuniones regulares para mantener la cohesión y la claridad en el progreso del proyecto.

Herramientas para Comunicaciones en el Equipo de Trabajo:

La comunicación interna del equipo se facilitará mediante plataformas colaborativas como Slack para mensajes instantáneos y discusiones, y herramientas como Trello

para el seguimiento visual de tareas. Se fomentará la transparencia y la apertura en la comunicación para mejorar la eficiencia y la comprensión mutua.

Relaciones con el Cliente (Entrevistas, Reuniones, Revisiones):

- Se llevarán a cabo entrevistas iniciales con los representantes del Colegio San Rafael para comprender sus necesidades. Reuniones regulares de revisión serán programadas para recibir comentarios continuos. La participación del cliente será esencial en todas las etapas para garantizar la alineación con sus expectativas.

Estándares de Documentación:

- La documentación seguirá estándares claros y uniformes, utilizando formatos accesibles y comprensibles para todos los miembros del equipo. Se establecerán protocolos para la actualización constante de la documentación a medida que evoluciona el proyecto.

Estándares de Código:

- El equipo seguirá prácticas de codificación consistentes con las directrices establecidas por la comunidad de desarrollo. Se realizarán revisiones de código periódicas para garantizar la calidad y coherencia del código, siguiendo estándares específicos de la industria.

Medición del Desempeño o Progreso:

- El progreso del proyecto se medirá mediante hitos y entregables establecidos en el cronograma. Se utilizarán métricas de rendimiento para evaluar la eficiencia del equipo y la calidad del producto. Revisiones regulares y retrospectivas contribuirán a la mejora continua.

Plan de Gestión de Cambios:

- Se implementará un plan detallado para gestionar cambios en requisitos, plazos o recursos. Los cambios se evaluarán en función de su impacto y se comunicarán de manera efectiva a todos los miembros del equipo y al cliente.

Control de Versiones (Método y Herramientas):

- El control de versiones se llevará a cabo utilizando un sistema basado en Git, con ramas para desarrollo, pruebas y producción. Se seguirá un modelo de integración continua para garantizar la estabilidad y la trazabilidad del código.

Gestión de Calidad (Proceso y Herramientas):

- El proceso de gestión de calidad incluirá pruebas exhaustivas en todas las etapas del desarrollo. Se utilizarán herramientas como Jira para el seguimiento de problemas y la mejora continua del proceso de calidad. La retroalimentación del cliente también será esencial para la evaluación de la calidad del producto final.