

Universidad de Pinar del Rio “Hermanos Saiz Montes De Oca”

Facultad: Ciencias Técnicas

Sistema de Informatización para la Universidad Pinar del Río

Proyecto Final de Ingeniería

de Software 3

Claudia de Jesús Santiesteban Bravo

Laura Elena Mesa

Carlos Alfredo Pérez Monterrey

Andrés Ernesto Luis Pastrana

Ray Antonio Francisco Muñes

Pinar del Rio, Cuba

2022

Índice de contenido

INTRODUCCION 3

GESTION DEL PROYECTO 4

Asignación de roles y responsabilidades al equipo de desarrollo 4

Evaluación y autoevaluación del desempeño del rol 4

Describir los problemas presentados a lo largo del desarrollo, como se solucionaron y las lecciones aprendidas 5

ANALISIS 5

Descripción General del Negocio 5

Diagrama de Caso de Uso Negocio ………………………………………………………………………………………6

Reglas de Uso del Negocio 6

ARQUITECTURA 7

Arquitectura candidata 7

Modelo de despliegue: 10

Estructuración en capas del sistema: 10

DISEÑADOR 12

Clases utilizadas sus atributos y descripción 12

Diseño de las Interfaces 14

ESPECIALISTA DE CALIDAD 17

Lista de chequeo para los artefactos 17

GESTOR DE CONFIGURACION 20

Elementos de configuración 20

CONCLUSIONES 21

**Introducción**

El trabajo de equipo es complejo y en ocasiones puede estar lleno de problemas de gestión de calidad, de planificación del equipo, desempeño de cada rol individualmente y en equipo. En este trabajo queremos solucionar todos los problemas que se pueden presentar sobre la marcha y la realización de las actividades propuestas para cada rol

**Gestión de Proyecto:**

**Asignación de roles y responsabilidades al equipo de desarrollo**

|  |  |
| --- | --- |
| Nombre y Apellido | Roles |
| Ray Antonio Francisco Núñez | Jefe de Proyecto y Analista |
| Carlos Alfredo Pérez Monterrey | Arquitecto y Programador |
| Andrés Ernesto Luis | Planificador |
| Laura Elena Mesa | Diseñador y Gestor de Configuración |
| Claudia de Jesús Santiesteban Bravo | Especialista de Calidad |

Tabla 1 Asignación de roles a los miembros d equipo

**Evaluación y autoevaluación del desempeño del rol así como el aporte de cada miembro del equipo al proyecto**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | JP | GC | Arq | Pg | Anl | Dis | Pln | EC |
| JP | **5** | **4** | **5** | **4** | **5** | **5** | **4** | **5** |
| GC | **5** | **5** | **5** | **5** | **5** | **5** | **5** | **5** |
| Arq | **4** | **5** | **4** | **5** | **4** | **5** | **5** | **5** |
| Pg | **5** | **4** | **5** | **4** | **5** | **4** | **4** | **4** |
| Anl | **4** | **5** | **4** | **5** | **4** | **5** | **5** | **4** |
| Dis | **4** | **5** | **5** | **5** | **4** | **5** | **4** | **5** |
| Pln | **5** | **5** | **5** | **4** | **5** | **4** | **4** | **5** |
| EC | **4** | **5** | **5** | **5** | **4** | **5** | **4** | **5** |

Tabla 2 Evaluación y autoevaluación de los miembros del equipo en el desempeño de los roles asignados

Leyenda:

JP Jefe de Proyecto

GC Gestor de Configuración

Arq Arquitecto

Pg Programador

Anl Analista

Dis Diseñador

Pln Planificador

EC Especialista de Calidad

**Describir los problemas presentados a lo largo del desarrollo, como se solucionaron y las lecciones aprendidas**

|  |  |  |
| --- | --- | --- |
| Problema | Solución | Lecciones aprendidas |
| Integrar funcionalidades desarrolladas por diferentes miembros del equipo a la línea base del código fuente resulta complejo. | Uso del Git (plugin V Code) para controlar las versiones del código fuente. | Controlar las versiones del código fuente agiliza la detección y corrección de defectos del software. |
| Falta de comprensión por parte de algunos miembros del equipo de las tareas asignadas. | Elaboración de documentos que explican de manera detallada las tareas asignadas a los miembros del equipo. Se indican y distribuyen en la medida de lo posible los recursos necesarios (bibliografía, software, etc.) para cumplir con las tareas asignadas. | Al elaborar un documento con las tareas a cumplir por parte de cada miembro del equipo las tareas son mejor comprendidas. |
| Mejorar el tiempo límite de las tareas dentro de la iteración | Atomizar las tareas al punto de no superar las 48h para realizarse | Aprendimos que las tareas de mayor duración en algunos casos se cumplen antes de tiempo por lo que surge una mala optimización de la duración general del proyecto |

Tabla 3 Propuestas de mejora

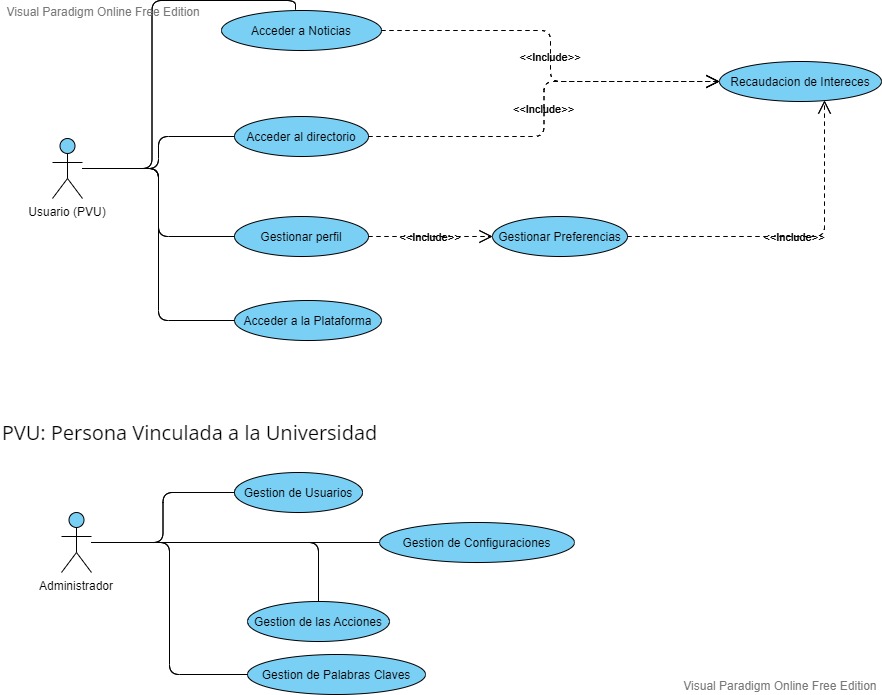
Análisis:

**Descripción General del Negocio**

UPR-k es un proyecto que sueña con esa Universidad que quepa dentro de un teléfono. Una aplicación con la que podamos conocer sobre su historia, su gente, sus áreas y espacios, una aplicación que nos permita tener un vínculo más estrecho con nuestra institución educativa. La propuesta pretende mostrar el quehacer de nuestras organizaciones y nuestra comunidad, mostrando las noticias y acontecimientos más recientes de manera más veloz e inmediata, donde el Sitial de Honor de las grandes personalidades y rectores que hemos tenido tome otra dimensión. Una APK que muestre de dónde venimos, quiénes somos y hacia dónde vamos, resaltando las potencialidades de nuestro claustro, los deseos de los educandos y los logros obtenidos por los egresados.

UPR-k aspira a que la biblioteca salga de sus salones y llegue hasta tu móvil, para que todos puedan acceder a sus servicios y, por qué no, solicitar nuevos para de esa manera, crecer.

**Diagrama de Caso de Uso Negocio**



**Reglas de Uso del Negocio:**

* El Usuario debe estar autenticado para acceder a la plataforma
* El Usuario debe contar con una cuenta activa en la plataforma SIGENU de la UPR
* Cada noticia, archivo, video, evento y contenido con el que se pueda interactuar estará vinculado con al menos una palabra clave
* Las palabras claves son únicas e irrepetibles
* Cada acción ejecutada por el usuario sobre un contenido X tendrá una puntuación pre configurada y se almacenara en relación con las palabras claves asociadas a el contenido
* Todo usuario debe tener una configuración mínima de intereses para el correcto funcionamiento del sistema de recomendación

**Arquitectura**

**Arquitectura candidata:**

|  |  |
| --- | --- |
| **Nombre Herramienta** | **Clasificación** |
| Visual Studio Code | IDE de desarrollo |
| Axure RP | Herramienta de diseño |
| Jira | Herramienta de planificación de proyectos |

**¿Por qué usar Axure RP?**

Axure RP es una de las herramientas indispensables en formaciones en Diseño UX, es decir, en diseño orientado a la experiencia de usuario. La propia definición de esta herramienta que encontramos en su página web ya nos aporta una visión de su enorme utilidad: “Axure RP pone el poder del prototipado en manos de gerentes de productos, analistas de negocios y profesionales de la experiencia de usuario. Antes de escribir una sola línea de código, las soluciones pueden ser prototipadas y validadas por las personas que mejor entienden su negocio, productos y clientes.”

Axure RP es un software o programa informático compatible tanto con el sistema operativo Windows de Microsoft como con macOS de Apple cuya principal utilidad es el diseño de sitemaps, wireframes y prototipos básicos o avanzados para proyectos de diseño UX.

Entre las principales ventajas de Axure RP destaca, en primer lugar, que se trata de una herramienta muy completa y a la vez, fácil de utilizar. En función de los conocimientos y de la propia experiencia del diseñador, éste podrá ir incorporando más y más funcionalidades que simularán todas las interacciones propuestas en su proyecto.

**¿Por qué usar Visual Studio Code?**

**Multiplataforma:** Es una característica importante en cualquier aplicación y más si se trata de desarrollo. Visual Studio Code está disponible para Windows, GNU/Linux y macOS.

**IntelliSense**: Esta característica está relacionada con la edición de código, autocompletado y resaltado de sintaxis, lo que permite ser más ágil a la hora de escribir código. Como su nombre lo indica, proporciona sugerencias de código y terminaciones inteligentes en base a los tipos de variables, funciones, etc. Con la ayuda de extensiones se puede personalizar y conseguir un IntelliSense más completo para cualquier lenguaje.

**Depuración:** Visual Studio Code incluye la función de depuración que ayuda a detectar errores en el código. De esta manera, nos evitamos tener que revisar línea por línea a puro ojo humano para encontrar errores. VS Code también es capaz de detectar pequeños errores de forma automática antes de ejecutar el código o la depuración como tal.

**Uso del control de versiones**: Visual Studio Code tiene compatibilidad con Git, por lo que puedes revisar diferencias o lo que conocemos con git diff, organizar archivos, realizar commits desde el editor, y hacer push y pull desde cualquier servicio de gestión de código fuente (SMC). Los demás SMC están disponibles por medio de extensiones.

**Extensiones:** Hasta ahora, he mencionado varias veces el término *extensiones* porque es uno de los puntos fuertes. Visual Studio Code es un editor potente y en gran parte por las extensiones. Las extensiones nos permiten personalizar y agregar funcionalidad adicional de forma modular y aislada. Por ejemplo, para programar en diferentes lenguajes, agregar nuevos temas al editor, y conectar con otros servicios. Realmente las extensiones nos permiten tener una mejor experiencia, y lo más importante, no afectan en el rendimiento del editor, ya que se ejecutan en procesos independientes.

**¿Por qué usar Jira?**

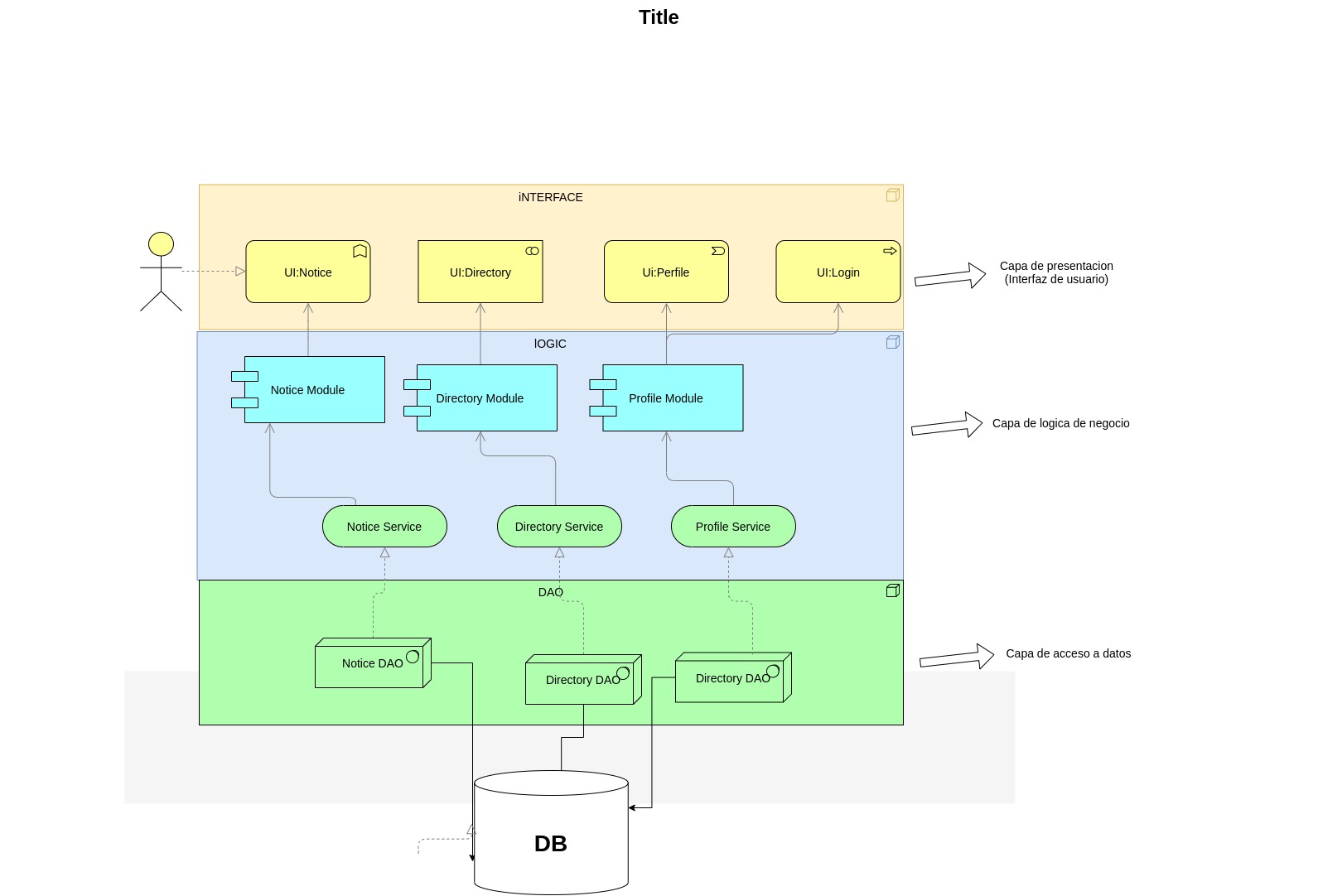
Jira Software proporciona herramientas de planificación y hojas de ruta para que los equipos puedan gestionar a los interesados, los presupuestos y los requisitos de las funciones desde el primer día.

Con el transcurso del tiempo, Jira ha buscado servir para cualquier tipo de gestión de proyectos ya bien sea el desarrollo de un sistema, llevando el seguimiento a los errores que vayan surgiendo durante el proceso de creación del software, gestión de contenidos, planificación de tareas de marketing, gestión de proyectos ágiles (Scrum), gestión de incidencias, entre otras.

## **Ventajas de Jira**

* A continuación, te listamos algunas de las ventajas de Jira:
* Permite trabajar con una planificación flexible
* Es posible implementar Kanban, Scrum o metodologías mixtas
* Elaboración de informes
* Intuitivo y de fácil manejo
* Jira es ideal para equipos de hasta diez miembros
* Herramienta en línea, cuenta con app

**Estructuración en capas del sistema:**



**Modelo de despliegue:**

Cliente

PC

Móvil

Tablet

Servidor

TCP/IP

|  |  |
| --- | --- |
| **Nodo** | **Descripción** |
| **PC**  **Dispositivo móvil**  **Tablet** | - Contiene el jar de la aplicación.  - Computadoras, teléfonos móviles y Tablet donde se va a ejecutar la aplicación. En este caso serían los dispositivos  las de los usuarios |
| **Servidor** | - Contiene el servidor de BD y el servidor de aplicaciones. - El servidor de almacenamiento o de BD, hace referencia en este caso a la ubicación del gestor de bases de datos. - El servidor de aplicaciones hace referencia a todos los elementos necesarios para correr la aplicación. |

**Front-End:**

**Ionic framework** : Permite desarrollar y desplegar aplicaciones híbridas, que funcionan en múltiples plataformas, como iOS nativo, Android, escritorio y la web (como una aplicación web progresiva), todo ello con una única base de código.

Ofrece un diseño limpio, sencillo y funcional.

**Angular framework:** Coherencia y reutilización del código. Tiene un buen soporte de Google y excelente soporte de la comunidad

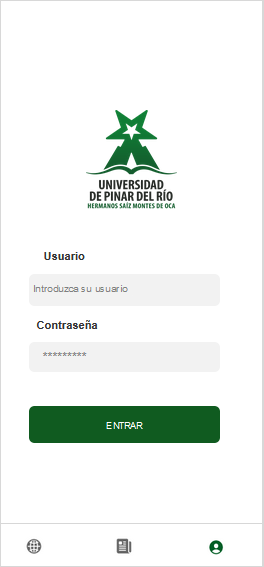
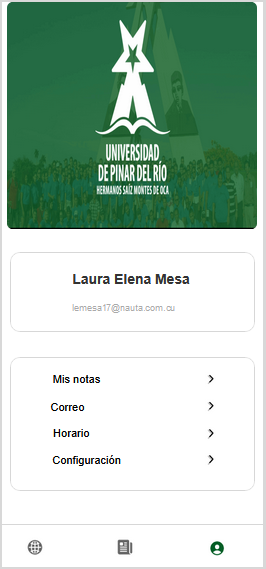
**Typescript:** El código de TypeScript es más fiable y simple de refactorizar. Intercambiabilidad. Puedes usar todas las bibliotecas y el código de JavaScript que quieras en tu código de TypeScript. Además, puedes adoptar gradualmente TypeScript en tu base de código de JavaScript.

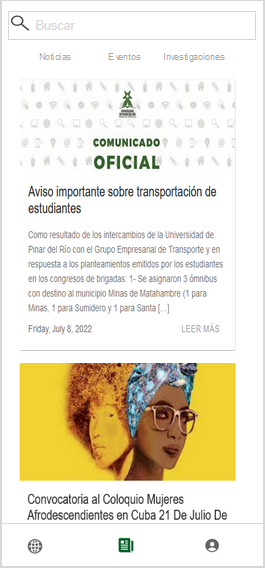
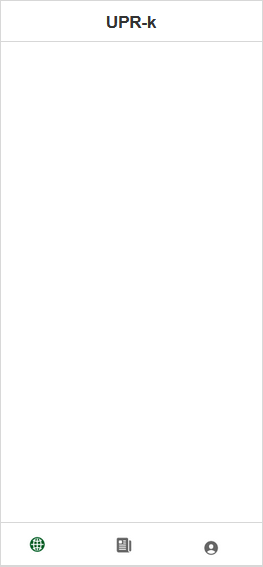
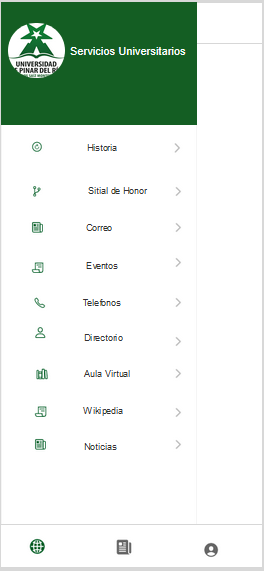
**Diseñador**

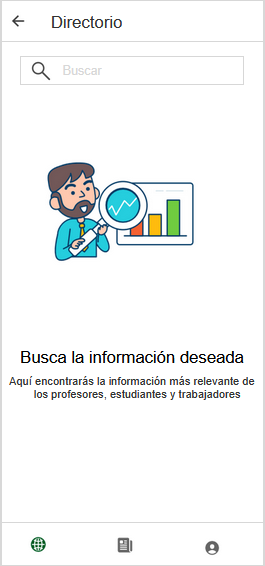
**Clases utilizadas sus atributos y descripción**

|  |  |  |  |
| --- | --- | --- | --- |
| Clases | Descripción | Atributos | Descripción |
| Notice | Esta clase se utiliza para tipar la respuesta que se recibe de la API al hacer la petición para mostrar las noticias. | id | Identificador único para la noticia. |
| date | La fecha en que se publicó la noticia, en la zona horaria del sitio. |
| date\_gmt | La fecha en que se publicó la noticia, como GMT. |
| guid | El identificador único global de la noticia. |
| modified | La fecha en que se modificó la noticia por última vez, en la zona horaria del sitio. |
| modified\_gmt | La fecha en que se modificó la noticia por última vez, como GMT. |
| slug | Un identificador alfanumérico para la noticia exclusivo de su tipo. |
| status | Un estado con nombre para la noticia. |
| type | Tipo de publicación para la noticia. |
| link | URL la noticia. |
| title | El título de la noticia. |
| content | El contenido de la noticia |
| excerpt | El extracto de la noticia. |
| autor | El identificador del autor de la noticia. |
| featured\_media | El ID de los medios destacados para la noticia. |
| coment\_status | Si los comentarios están abiertos o no en la noticia. |
| template | El archivo de tema que se usará para mostrar la noticia. |
| format | El formato de la noticia |
| meta: | Metacampos. |
| categories | Los términos asignados a la noticia en la taxonomía de categorías |
| tags | Los términos asignados a la noticia en la taxonomía post\_tag. |
| DirectoryUser | esta clase se utiliza para tipar la respuesta que se recibe de la API al hacer la petición para buscar en el directorio la información de los trabajadores y estudiantes de la Universidad | status | el estado de la respuesta |
| message | mensaje de la respuesta, si fue exitosa o no |
| data | un arreglo de personas |
| Person | se utiliza para acceder a los datos de la persona que se está buscando en el directorio | displayname | objeto con el nombre de la persona completa |
| physicaldeliveryofficename | facultad en donde está o si es baja |
| mail | correo de la persona |
| samaccountname | nombre de usuario |

**Diseño de las Interfaces**

** **

****  ** **  ****

****

**Especialista de Calidad**

Lista de chequeo para los artefactos

|  |  |  |
| --- | --- | --- |
| Artefacto: | Plan del proyecto | |
| Pregunta | Cumple | No cumple |
| ¿En este artefacto se encuentran contempladas todas las tareas necesarias para realizar el proyecto? |  | X |
| ¿El tiempo que se asignó a cada tarea corresponde con la complejidad de esta para su realización? |  | X |
| ¿Corresponde cada tarea asignada a las capacidades de cada miembro del equipo? | X |  |
| ¿Se utiliza algún sistema de Gestión de Proyecto? | X |  |
| ¿Están presente todas las tareas que le corresponde al Jefe de Proyecto? |  | X |
| ¿Están presente todas las tareas que le corresponde al Analista? | X |  |
| ¿Están presente todas las tareas que le corresponde al Arquitecto? | X |  |
| ¿Están presente todas las tareas que le corresponde al Gestor de Configuración? | X |  |
| ¿Están presente todas las tareas que le corresponde al Programador? | X |  |
| ¿Están presente todas las tareas que le corresponde al Diseñador? | X |  |
| ¿Están presente todas las tareas que le corresponde al Planificador? |  | X |
| ¿Están presente todas las tareas que le corresponde al Especialista de Calidad? | X |  |
| ¿Todas las tareas poseen los recursos y la fecha? |  | X |

|  |  |  |
| --- | --- | --- |
| Artefacto: | Diagrama Caso de Uso de Negocio | |
| Pregunta | Cumple | No cumple |
| ¿Apresen todos los actores que pertenecen al negocio? | X |  |
| ¿Asociaron correctamente los casos de uso a los actores que pertenecen? | X |  |
| ¿El diagrama describe correctamente la finalidad y el objetivo del negocio? | X |  |
| ¿El nombre de cada Caso de Uso se corresponde con la descripción del mismo? | X |  |

|  |  |  |
| --- | --- | --- |
| Artefacto: | Reglas de Negocio | |
| Pregunta | Cumple | No cumple |
| ¿Se elaboraron correctamente las reglas del negocio? | X |  |
| ¿Tiene las reglas del negocio una clara interpretación y significado? | X |  |
| ¿Las reglas de negocio identificadas se cumplen en todo momento? |  | X |

|  |  |  |
| --- | --- | --- |
| Artefacto: | Vista de la Arquitectura estructurada | |
| Pregunta | Cumple | No cumple |
| ¿Están bien definidos los límites de las capas de la arquitectura? | X |  |
| ¿Cada uno de los componentes de las capas está bien explicados? | X |  |
| ¿Cada componente está correctamente ubicado? | X |  |

|  |  |  |
| --- | --- | --- |
| Artefacto: | Modelo de despliegue | |
| Pregunta | Cumple | No cumple |
| ¿Se definieron correctamente todos los nodos? | X |  |
| ¿La comunicación entre los nodos está identificada? | X |  |
| ¿Los componentes identificados son todos existentes? | X |  |

|  |  |  |
| --- | --- | --- |
| Artefacto: | Estándar de codificación | |
| Pregunta | Cumple | No cumple |
| ¿Todas las clases definidas cumplen correctamente el estándar de codificación? | X |  |
| ¿Las variables definidas en el proyecto siguen dicho estándar? |  | X |
| ¿Todas las interfaces definidas cumplen correctamente el estándar de codificación? | X |  |
| ¿Todos los métodos están comentados? | X |  |
| ¿Si el valor de una variable no cambia se define como una constante? |  | X |

**Gestor de Configuración**

**Elementos de configuración:**

1- El informe del proyecto

2- La ficha de gestión de proyecto

3- El estándar de codificación utilizado

**Conclusiones**

UPR-k es un proyecto que impulsa al equipo a trabajar y avanzar día a día con la realización de este trabajo nos dimos cuenta de que un equipo no es perfecto ya que hemos cometido errores de planificación de las actividades, a la hora de realizar algunos artefactos, entre otros pero el objetivo más importante es saber trabajar en equipo aprender de nuestros errores siendo críticos y autocríticos y creo que eso lo hemos logrado juntos como un gran equipo de desarrollo.

**Bibliografía**

1. Ionic Documentation

2. Angular Documentation