**SOLICITUD DE PRÁCTICA EMPRESARIAL**

**I.- Identificación del Estudiante**

**NOMBRE:** ALVARO DE JESUS FRIAS GAMEZ

**CÉDULA DE CIUDADANÍA:** 73.107.111

**DIRECCIÓN:** Calle 11 # 30 -57 Barrio Villalba - Valledupar

**CELULAR:** 315 559 4077

**CORREO ELECTRÓNICO:** aljesuf@gmail.com

**II.- Identificación De La Empresa/Proyecto De Investigación**

**EMPRESA/PROYECTO**:

UPC-System S.A.S

**DIRECCIÓN:**

CL 16 B # 19 A 63. Centro Comercial Las Américas – Cartagena de Indias

**TELÉFONO:**

**315 551 7989**

**NIT:**

9001234587-8

**REPRESENTANTE LEGAL:**

CARMEN ALICIA SOLANO GUERRERO

**CIUDAD:**

Cartagena de Indias

**JEFE INMEDIATO: (Nombre y Cargo):**

Tomás Barrios González

Representante legal

**AREA DE TRABAJO:**

Desarrollo de software

**III.- Problema a Solucionar**

**UPC-System S.A.S** es una empresa que se encarga de utilizar la tecnología para mejorar y automatizar una gran variedad de negocios brindando servicios de software a la medida y soporte a dichos productos, dependiendo de la necesidad del cliente; para ello hace uso de las tecnologías de la información y las comunicaciones.

En las labores de soporte técnico en la empresa **UPC-System S.A.S** se detectó la necesidad de garantizar que tanto el producto final como cada uno de los módulos desarrollados se acercan a la perfección en cuanto a calidad, seguridad y confiabilidad. Para garantizar esta calidad existe un proceso en el desarrollo software, tras la realización del análisis, diseño y en algún punto del desarrollo de la aplicación se inicia la indispensable etapa de pruebas, sin embargo, no se cuenta con un equipo de **Testing**, ni se aplican ninguno de los tipos de pruebas de software, lo cual ha generado muchos inconvenientes en los productos ofrecidos:

* Se consume demasiado tiempo detectado y corrigiendo errores detectados en la etapa de producción.
* Integrar nuevas funcionalidades en ocasiones genera errores en módulos que son causantes de pérdidas económicas para los clientes y perdida de confiabilidad en la empresa.
* La utilización de la aplicación en distintos escenarios como navegadores o dispositivos generan un comportamiento distinto que puede causar una mala utilización del software.
* Una vez un modulo nuevo ha sido desarrollado no hay forma de asegurarse que su funcionamiento cumple con cada uno los requerimientos y que su comportamiento seguirá siendo el adecuado cuando se realicen modificaciones en el código.

Estos problemas pueden ser solventados desarrollando un ***framework****,* **entorno de trabajo**​ o **marco de trabajo,** el cual contiene un conjunto de conceptos, guías y herramientas enfocadas a resolver la problemática relacionadas con el **Testing** del software el cual incluiría herramientas para facilitar la corrección de errores donde sean detectados con la aplicación de los principios y patrones de diseño de software, siempre con el objetivo de mejorar la calidad, rendimiento y seguridad de los productos de software ofrecidos. Para el desarrollo de las pruebas es necesario un ambiente aislado del de desarrollo, es decir, deberán ejecutarse en un entorno idéntico al de producción las cuales son: Amazon Web Services (AWS abreviado) el cual es una colección de servicios de computación en la nube pública ofrecidas a través de Internet por Amazon.com, PostgreSQL un sistema de gestión de bases de datos relacional y *Python* un lenguaje de programación interpretado y multiparadigma.

A continuación, una breve descripción de cada de algunas pruebas que se necesitan estandarizar en el marco de trabajo y de las guías a desarrollar para realizar nuevas pruebas:

* **Pruebas Unitarias o de Componente:** Permitirán verificar que los componentes unitarios están codificados bajo condiciones de robustez, esto es, soportando el ingreso de datos erróneos o inesperados y demostrando así la capacidad de tratar errores de manera controlada. Es importante que toda la funcionalidad de cada componente unitario sea cubierta, por al menos, dos casos de prueba, los cuales deben centrarse en probar al menos una funcionalidad positiva y una negativa.

También es importante contar con estas pruebas en todos y cada uno de los módulos por mencionar algunos están: facturación, inventario, control de activos, control de caja, control de banco, administrador de bodegas, manejo de cartera, producción, entre otros.

* **Pruebas de Integración:** Estas pruebas se hacen necesarias para garantizar que elementos del software que interactúan entre sí, funcionan de manera correcta y que al integrar nuevas funcionalidades el sistema (módulos en desarrollo, por ejemplo: Contabilidad, modulo offline, aplicación móvil) se tendrá la menor cantidad de errores posibles.
* **Pruebas de Sistema:** Estas pruebas permitirán verificar que la funcionalidad total del sistema fue implementada de acuerdo a los documentos de especificación definidos. Los casos de prueba a diseñar en este nivel de pruebas, deben cubrirán los aspectos funcionales y no funcionales del sistema. Para el diseño de los casos de prueba en este nivel, se realizará como bases de prueba entregables tales como: requerimientos iniciales, historias de usuario, diseños, casos de uso, manuales técnicos y manuales de usuario para los módulos de más importancia. Por último, como se menciono anteriormente las pruebas desarrolladas en este nivel se desplegarán en un ambiente similar al de producción, evitando en todos los casos utilizar el ambiente real de los clientes, debido principalmente, a que pueda ocasionar fallos en los servidores o bloqueos en la base de datos lo que ocasionaría indisponibilidad los ofrecidos.
* **Pruebas de Aceptación:** Permitirán verificar que el sistema funciona y lo hace de acuerdo con las especificaciones y necesidades de los usuarios finales. En este nivel será necesario contar con el apoyo de usuarios finales y en lo posible ejecutar las pruebas junto con el cliente, entregándoles conocimientos de primera mano sobre la funcionalidad del sistema y garantizándole el correcto funcional del módulo respectivo.

El marco de trabajo – ***framework -*** contendrá información técnica de las pruebas y de cómo realizar aplicarlas a nuevos módulos de la forma más optima posible con el objetivo de hacerlas más robustas, fáciles de mantener y con código mas estandarizado y entendible para otros programadores. Todo esto aplicando en lo posible los principios de programación SOLID, control de cohesión y acoplamiento, reutilización de código y patrones de diseño de software.

La solución al problema identificado requiere unos amplios conocimientos en tecnologías de administración de servidores, bases de datos, programación y conceptos de ingeniería de software por esto solicito que sean aceptadas como practicas formativas para optar al título de **Ingeniero De Sistemas**. Estas actividades no se cruzarían con el objetivo del contrato ya vigente el cual se enfoca en apoyo de instalación y soporte para usuarios finales.

**IV.- Cronograma de Actividades**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| MESES | Marzo | Abril | Mayo | Junio |
| PLANEACIÓN |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| ANÁLISIS: |  |  |  |  |
| Estudio Sistema Actual |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Requerimientos |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Casos de uso e historias de usuario para módulos a testear |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| TESTING: |  |  |  |  |  |  |  |
| Configuración del entorno |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Diseño global de pruebas de componentes |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Pruebas unitarias para cada uno de los modelos (entidades) |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  Pruebas de integración |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Pruebas de sistema  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  Pruebas de aceptación |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| CAPACITACIÓN |  |  |  |  |  |  |  |
| MANUAL DE USUARIO |  |  |  |  |  |  |  |
| IMPLEMENTACION |  |  |  |  |  |  |  |

Para la realización de estas actividades se invertirán 24 horas semanales durante 14 semanas, aproximadamente 336 horas de trabajo.

**V.- Anexos (Todos los indicados en la “Guía de Práctica Empresarial”)**