



Las Americas Institute of Technology

Nombres:

Misael Ramírez 2020-9910

Período:

Septiembre-diciembre

Materia:

Programación III

Profesor:

Kelyn Tejada

Práctica:

Proyecto Final

Índice

Metodología de Trabajo Agile-Scrum	2
Requisitos iniciales.....	Error! Bookmark not defined.
Tecnología para aplicar:	Error! Bookmark not defined.
Objetivo del proyecto:	Error! Bookmark not defined.
Alcance del proyecto:	Error! Bookmark not defined.
Definición de lo que el sistema en un primer Reléase va a poder hacer:.....	Error! Bookmark not defined.
Definición de requerimientos del sistema para el primer Release	2
Equipo Metodología Scrum	3
Definición de tareas a ejecutar:	3
Definición del equipo de trabajo (roles, habilidades, etc.):.....	3
Herramientas por utilizar:.....	4
Definición de épicas:.....	Error! Bookmark not defined.
Acciones:.....	Error! Bookmark not defined.
Definición de fechas para cada ceremonia de Scrum.....	Error! Bookmark not defined.
Historias de usuarios con sus puntos y criterios de asignación	5
Demostración.....	5
Plan de pruebas	5
Lista de requerimientos funcionales y no funcionales de acorde a las historias de usuarios	Error! Bookmark not defined.
Herramientas de pruebas que se estarán usando	6
Definir cuáles son los criterios de aceptación de las pruebas.....	6
Definir cuáles son los criterios de rechazo en las pruebas.....	Error! Bookmark not defined.
Estimación del tiempo de la ejecución de pruebas. (elaborar cronograma de trabajo)	7
Plantillas para caso de pruebas	8
Plantilla con los equipos de pruebas y sus responsabilidades	9
Plan de automatización de pruebas	10
Ejecución y demostración del plan de automatización de pruebas	Error! Bookmark not defined.
Bibliografía	12

Metodología de Trabajo Agile-Scrum

Selector de Materias

Requisitos iniciales

Tecnología por utilizar:

En este proyecto relacionado con las metodologías de trabajo Agile-Scrum, se empleará el lenguaje de programación JavaScript.

Objetivo del proyecto:

El propósito principal de este proyecto es desarrollar una página web que permita realizar la selección de materias, aplicando la metodología Agile-Scrum para optimizar el desarrollo y la gestión del proyecto.

Alcance del proyecto:

El objetivo a largo plazo de este proyecto es ser implementado en instituciones educativas, considerando que la virtualidad será una realidad creciente en las escuelas y colegios de la República Dominicana en los próximos años.

Definición del alcance funcional en el primer Release:

En su versión inicial, el sistema contará con varias funciones esenciales que incluirán un selector de materias con características clave como:

- Registrar los datos del estudiante
- Carrera para elegir del estudiante, en esta opción encontraras:
- Software
- Mecatrónica y redes
- Redes
- Selección de los horarios de clase
- Ver el horario seleccionada por día y hora

Definición de requerimientos del sistema para el primer Release.

En esta etapa inicial, el sistema requerirá la implementación de diversos métodos de decodificación para su desarrollo, tales como: Creación de la estructura principal de la página:

- Separar las vistas.
- Creación de los campos.
- Añadirle la parte visual con Bootstrap.

Equipo Metodología Scrum

Definición de tareas a ejecutar:

Las tareas por ejecutar que tendrá nuestro selector de materias van de la mano con los requerimientos establecidos, estas tareas serán las siguientes:

- **Creación de las vistas principales:** esta parte es para ir definiendo los principales campos que necesitaremos.
- **Creación de elementos:** En esta parte se crearán los campos, labels, botones, etc. Los cuales se utilizarán para el manejo efectivo de la página según los requerimientos.
- **Darle los arreglos visuales:** En este paso iremos mejorando la parte visual de nuestra página web para darle a los usuarios una mejor experiencia de uso.
- **Correctos funcionamientos de los elementos:** validar que cada uno de los campos y botones funcionen correctamente.
- **Realizar validaciones:** realizar la validación para que los usuarios no puedan dejar campos vacíos a la hora de registrarse.

Definición del equipo de trabajo (roles, habilidades, etc.):

Scrum Máster:

- Misael Ramírez Pérez

Va a hacer la persona de tiempo completo, facilitador o (Coach) de Scrum.

PO (Product Owner):

- Jorge Linares

Será la persona responsable de asegurar que el equipo aporte valor al negocio, y apoyar las necesidades de todos los usuarios en el negocio, así como también las necesidades y el funcionamiento del Equipo Scrum.

DEV (Developer):

- Diego Sánchez – DEV
- Sofía Astacio - DEV

Serán las personas expertas en escribir código, es decir, idear el conjunto de secuencias de órdenes que llevan a un sistema informático a realizar una acción concreta.

QA (Quality Assurance):

- Pedro Olivares - QA

Plantear la estrategia de pruebas. Diseñar los escenarios y casos de prueba. Revisar, planificar, preparar y ejecutar las pruebas. Reportar errores, plantear soluciones y validar la corrección.

Herramientas por utilizar:

Para este proyecto se estarán utilizando diferentes herramientas para la elaboración de este, entre ellas estarán:

- **HTML-5:** son códigos utilizados para "marcar" el texto de una página web, con el fin de dar instrucciones al navegador sobre cómo mostrarlo. Es decir, las etiquetas HTML son el lenguaje utilizado para estructurar y definir el contenido en un documento HTML.
- **JavaScript:** es un lenguaje de programación que los desarrolladores utilizan para hacer páginas web interactivas. Desde actualizar fuentes de redes sociales a mostrar animaciones y mapas interactivos, las funciones de JavaScript pueden mejorar la experiencia del usuario de un sitio web.
- **Bootstrap:** es una biblioteca multiplataforma o conjunto de herramientas de código abierto para diseño de sitios y aplicaciones web.

Definición de épicas

El sistema tiene como objetivo permitir a los usuarios llevar a cabo las funciones esenciales asociadas con un selector de materias.

Acciones:

Establecimiento de fechas para las ceremonias Scrum:

El equipo Scrum llevará a cabo un total de cuatro reuniones, distribuidas en un período de tres semanas, y programadas en las siguientes fechas:

○ Martes 10 de junio, 11:30 AM:

- Creación de Historias de usuario
- Asignación de puntuaciones de las historias de usuario
- Reasignación de fechas y reuniones.
- Realización de los criterios de aceptación.

○ Viernes 13 de junio, 5:00 PM

- Confirmación de las estimaciones de historias para el sprint planning.
- Evaluación e incorporación de nuevas historias o ajuste de historias.
- Se replantearán aquellas historias con altos puntos por historias más pequeñas.

○ Miércoles 18 de junio, 9:30 PM

- Inspección de los puntos débiles en las historias de usuarios.
- Realización del plan de prueba ➤ **Lunes**

23 de junio, 10:00 AM

- Chequeo del proyecto en general.

Historias de usuarios con sus puntos y criterios de asignación

Enlace de Azure:

<https://dev.azure.com/20209910/Subject%20Selector>

Demostración

<https://drive.google.com/file/d/1eo8fWBwYd1O-iCF79akqP3McVw1GQo3k/view?usp=sharing>

Plan de pruebas

Lista de Requerimientos Funcionales y No Funcionales Basados en las Historias de Usuario

1. Pruebas de funcionalidad básica:

- Comprobar que es posible crear nuevas cuentas de usuario y que los datos se almacenan correctamente en la base de datos.
- Verificar que los usuarios pueden iniciar sesión con cuentas previamente creadas.
- Validar que se pueden agregar datos predeterminados a las cuentas de usuario, como provincia y ciudad.
- Asegurar que los campos de registro se desplieguen adecuadamente en la vista principal de la aplicación.
- Garantizar que los botones realizan las acciones correspondientes y funcionan correctamente.
- Confirmar que las animaciones se reproducen sin afectar la funcionalidad de la aplicación.
- Verificar que los campos se posicionan correctamente en la vista final de la aplicación.
- Validar que todos los campos requeridos se verifican adecuadamente y que se muestran mensajes de error en caso de datos incompletos o incorrectos.
- Asegurar que el menú de navegación para regresar a vistas anteriores opera correctamente sin comprometer la funcionalidad de la aplicación.

2. Pruebas de validación de campos:

- Verificar que los campos obligatorios, como nombre, provincia, ciudad y carrera, se validan correctamente.
- Asegurar que los campos de texto, como sector y calle, no permiten caracteres no válidos.
- Validar que las opciones de selección de carreras limitan la elección a un máximo de tres.
- Confirmar que las selecciones de horarios no permiten duplicados ni superposiciones.

3. Pruebas de funcionamiento:

- Comprobar que la aplicación es compatible con diferentes navegadores y dispositivos.
- Validar que la aplicación se carga y responde rápidamente, incluso cuando se maneja una gran cantidad de materias y horarios.
- Garantizar que la aplicación no se bloquee ni presente fallas durante su uso normal.
- Verificar que los datos se almacenen de manera correcta en la base de datos y puedan recuperarse exitosamente tras cerrar y volver a abrir la aplicación.

Herramientas de pruebas que se estarán usando.

En este sistema se estarán utilizando una herramienta para la elaboración de pruebas, las cuales son:

➤ Selenium IDE

Es un conjunto de utilidades que facilita la labor de obtener juegos de pruebas para aplicaciones web. Para ello nos permite grabar, editar y depurar casos de prueba, que podrán ser ejecutados de forma automática e iterativa posteriormente.

Definir cuáles son los criterios de aceptación de las pruebas

Validación de la información: se debe validar que se hayan completado todos los campos obligatorios antes de continuar con la selección de materias.

- Selección de carreras: se debe validar que el estudiante solo pueda seleccionar las 3 carreras disponibles y no se le permita seleccionar más.

- Selección de materias: se debe validar que el estudiante solo pueda seleccionar las materias correspondientes a la carrera que eligió y no se le permita seleccionar materias de otras carreras.
- Horarios de materias: se debe validar que cada materia tenga 3 horarios disponibles para que los estudiantes puedan elegir el más adecuado para ellos.
- Generación del horario: se debe validar que el horario generado no tenga conflictos de horario entre las materias seleccionadas.
- Mostrar información del estudiante: se debe validar que se muestren correctamente los datos del estudiante al finalizar el proceso de selección

Definición de los Criterios de Aceptación para las Pruebas

1. Validación de la información:

Se debe asegurar que todos los campos obligatorios estén correctamente completados antes de proceder con la selección de materias.

2. Selección de carreras:

El sistema debe garantizar que el estudiante solo pueda elegir entre las 3 carreras disponibles, sin permitir selecciones adicionales.

3. Selección de materias:

Se debe confirmar que el estudiante solo pueda elegir materias asociadas a la carrera seleccionada, restringiendo la opción de materias de otras carreras.

4. Horarios de materias:

Cada materia debe ofrecer 3 horarios distintos, y el sistema debe permitir que el estudiante seleccione el que mejor se ajuste a sus necesidades.

5. Generación del horario:

El sistema debe validar que el horario generado no presente conflictos entre las materias seleccionadas por el estudiante.

6. Visualización de información del estudiante:

Se debe verificar que los datos del estudiante se muestren correctamente al finalizar el proceso de selección.

Estimación del tiempo de la ejecución de pruebas. (elaborar cronograma de trabajo).

	Octubre			
PRUEBAS	SEMANA 1	SEMANA 2	SEMANA 3	SEMANA 4
Creación de las vistas principales				

Creación los campos de registro				
Agregar los datos por defecto				
Creación de los botones				
Añadirle las animaciones				
Crear las conexiones para que los campos se coloquen en la última vista				
Validar que todos los campos estén correctamente				
Hacer un menú para volver a las vistas anteriores				
Hacer validaciones de campos				
Hacer pruebas de funcionamiento				

Plantillas para caso de pruebas

PLANTILLA PARA CASOS DE PRUEBA	
PRUEBAS	CASO DE PRUEBA
Creación de las vistas principales	Validar que se cree según lo requerido
	Que estén los campos necesarios
Creación los campos de registro	Crear los campos del formulario de registro
	Crear los títulos
Agregar los datos por defecto	Agregar las informaciones que vienen por defecto al cargar la página.
Creación de los botones	Crear los botones de registro
	Crear botón de guardar
Añadirle las animaciones	Crear animaciones de alerta
	Crear animaciones a botones
Crear las conexiones para que los campos se coloquen en la última vista	Agregar las relaciones entre campos
	Agregar funcionamiento de botones
Validar que todos los campos estén correctamente	Validar que todo esté según lo requerido
Hacer un menú para volver a las vistas anteriores	Hacer menú para movernos entre vistas
Hacer validaciones de campos	Hacer validaciones y alertas en caso de error

Hacer pruebas de funcionamiento	Probar todo el funcionamiento
---------------------------------	-------------------------------

Plantilla con los equipos de pruebas y sus responsabilidades.

PLANTILLA PARA EQUIPOS DE PRUEBAS		
PRUEBAS	EQUIPOS DE PRUEBAS	RESPONSABILIDADES
Creación de las vistas principales	DEV	Desarrollar las vistas de la página.
Creación los campos de registro	DEV	Colocar todos los campos requeridos para la página.
Agregar los datos por defecto	DEV	Añadir todos los datos requeridos.
Creación de los botones	DEV	Diseñar los botones que ese utilizaran para el correcto funcionamiento.
Añadirle las animaciones	DEV	Agregar animaciones para que la pagina sea mas agradable para el usuario
Crear las conexiones para que los campos se coloquen en la última vista	DEV	Darle el correcto funcionamiento a los botones y formularios.
Validar que todos los campos estén correctamente	QA	Verificar que todos los campos funciones.
Hacer un menú para volver a las vistas anteriores	DEV	Crear un menú para poder movernos entre las vistas.
Hacer validaciones de campos	QA	Verificar que todos funcionen
Hacer pruebas de funcionamiento	QA	Validar que todo este según los requerimientos

Plan de automatización de pruebas

1. Identificación de flujos críticos:

Es esencial determinar los flujos más relevantes para el usuario, aquellos que requieren mayor atención durante las pruebas. En este caso, los flujos clave incluyen el registro del estudiante, la selección de carreras y materias, la validación y asignación de horarios, la generación del horario de clases, y las funcionalidades para guardar e imprimir el horario.

2. Selección de la herramienta de automatización:

Es necesario elegir una herramienta de automatización adecuada que cuente con las capacidades necesarias para cubrir los flujos críticos del proyecto. Opciones recomendadas incluyen Selenium, Cypress y Appium, dependiendo de los requisitos específicos.

3. Diseño de los casos de prueba:

Se deben crear casos de prueba detallados para cada flujo crítico, definiendo los pasos que el usuario debe seguir, los datos de entrada necesarios y los resultados esperados. Además, es importante incluir los criterios de aceptación y rechazo para cada caso de prueba.

4. Implementación de los casos de prueba:

Los casos de prueba deben ser implementados utilizando la herramienta de automatización seleccionada. Es fundamental que sean claros y comprensibles, facilitando su interpretación por otros miembros del equipo.

5. Ejecución de los casos de prueba:

Se procederá con la ejecución de los casos de prueba para evaluar el correcto funcionamiento de la página. Es imprescindible verificar que todos los flujos críticos se prueben adecuadamente y registrar los resultados de cada ejecución.

6. Generación de informes de resultados:

Se deben elaborar reportes que detallen el estado de las pruebas y los resultados obtenidos. Estos informes deben ser precisos y fáciles de interpretar, permitiendo que el equipo tome decisiones informadas basadas en ellos.

7. Mantenimiento de las pruebas:

Es importante mantener las pruebas actualizadas y alineadas con las modificaciones realizadas en la página. Esto incluye ajustar los casos existentes y añadir nuevos casos de prueba conforme surjan nuevas funcionalidades o cambios.

8. Integración con el desarrollo:

Las pruebas automatizadas deben integrarse en el flujo de desarrollo del proyecto, asegurando que todas las funcionalidades sean validadas antes de liberar nuevas versiones de la página.

Link Repositorio GitHub:

<https://github.com/Rubirosa/ProyectoFinal-3>

Link AzureDevOps:

<https://dev.azure.com/20209910/Subject%20Selector>

Bibliografía

- <https://www.softwaretestingbureau.com/crear-un-buen-plan-depruebas/#:~:text=El%20plan%20de%20pruebas%20tiene,cambios%20que%20tiene%20el%20proyecto>
- <https://spa.myservername.com/sample-test-plan-document>
- <https://docs.microsoft.com/en-us/azure/devops/test/perform-exploratorytests?view=azure-devops>
- https://es.wikipedia.org/wiki/Requisito_no_funcional
- <http://www.pmoinformatica.com/2015/05/requerimientos-no-funcionalesejemplos.html>
- <http://www.pmoinformatica.com/2017/02/requerimientos-funcionales-ejemplos.html>
- <https://scrum.menzinsky.com/2020/06/una-historia-de-usuario-puede-dividirse.html>
- <http://www.lecciones-aprendidas.info/2013/11/como-es-una-historia-de-usuarioun.html>
- <https://www.sinnaps.com/blog-gestion-proyectos/scrum-epica>
- <https://marketplace.visualstudio.com/items?itemName=ms.vss-exploratorytestingweb#supportedbrowsers>